

**24<sup>th</sup> International Conference on  
Information Modelling and  
Knowledge Bases**

**EJC 2014**

B. Thalheim

H. Jaakkola

Y. Kiyoki

June 3-6 2014, Kiel, Hotel Birke, Germany

# Impressum

## Editors

Bernhard Thalheim  
Christian-Albrechts-University Kiel  
Dept. of Computer Science  
Information Systems Engineering  
D-24098 Kiel  
Germany

Hannu Jaakkola  
Tampere University of Technology  
at Pori  
P.O.Box 300,  
FIN-28101 Pori  
Finland

Yasushi Kiyoki  
Graduate School of Media and Governance  
Keio University  
5322 Endoh  
Fujisawa, Kanagawa  
Japan, 252-0882

## Kiel Computer Science Series

The “Kiel Computer Science Series” (KCSS) is published by the Department of Computer Science of the Faculty of Engineering at Kiel University. The scope of this open access publication series includes conference proceedings, dissertation theses, habilitation theses, and text books in computer science.

Kiel Computer Science Series (KCSS) 2014/4 v1.0 dated 2014-05-13

ISSN 2193-6781 (print version)

ISSN 2194-6639 (electronic version)

Electronic version and errata available via <https://www.informatik.uni-kiel.de/kcss>

No updates after 2014-05-13

Published by the Department of Computer Science, Kiel University Information Systems Engineering

Please cite as: B. Thalheim, H. Jaakkola, Y. Kiyoki. Proceedings of the International Conference on Information Modelling and Knowledge Bases (EJC 2014), June 3-6, 2014, Kiel. Number 2014/2 in Kiel Computer Science Series. Department of Computer Science, Faculty of Engineering, Kiel University.

```
@PROCEEDINGS{EJC2014ConferenceProceedings,  
  TITLE =      {Proceedings of the International Conference  
                on Information Modelling and Knowledge Bases (EJC 2014)},  
  YEAR =      {2014},  
  editor =    {B. Thalheim, H. Jaakkola, Y. Kiyoki},  
  publisher = {Kiel University},  
  volume =   {2014/4},  
  series =    {KCSS},  
  organization = {Department of Computer Science, Faculty of Engineering},  
  month =     {June},  
  isbn =      {ISSN 2193-6781},  
  source =    {https://www.informatik.uni-kiel.de/kcss},  
}
```

©Bernhard Thalheim, Hannu Jaakkola, Yasushi Kiyoki

## Table of Contents

Database Structure Modeling by Stereotypes, Patterns, and Templates .....	1
<i>Bader AlBdaiwi, René Noack, and Bernhard Thalheim</i>	
GIS-Cloud Requirements Framework for E-Government Services .....	20
<i>Fahdah F. AlOthman, Weaam F. AlMazyad, and Ajantha Dahanayake</i>	
Trust levels of Mobile Banking Apps .....	28
<i>Ahlam Alshareed, Hadeel Alsagyyer, Hanieah Alenizi, and Ajantha Dahanayake</i>	
Conceptual Framework for Big Data Analytics Solutions .....	36
<i>Mashail Alswilmi, Nouf Alnajran, and Ajantha Dahanayake</i>	
Flexible Information Integration with Local Dominance .....	44
<i>Scott Britell, Lois Delcambre, and Paolo Atzeni</i>	
Modeling of classification error rate based on neural networks learners .....	63
<i>Boštjan Brumen, Ivan Rozman, and Aleš Černezel</i>	
Formation of a Collaborative Society .....	71
<i>Ladislav Burita and Vojtech Ondryhal</i>	
FOCAPLAS – A platform for cloud application development and running support .....	79
<i>Xing Chen and Keiichi Shiohara</i>	
Comparison of Measurements of Learner’s Performance .....	99
<i>Aleš Černezel and Boštjan Brumen</i>	
Event ontology specification based on the theory of valency frames .....	108
<i>Martina Čihalová</i>	
Musical Tunes Emotions Identification System by means of Intrinsic Musical Characteristics .....	123
<i>Tatiana Endrjukaite and Yasushi Kiyoki</i>	
Human Reaction in Thailand based on Social Media Analysis after the East Japan Great Earthquake .....	143
<i>Takako Hashimoto, Supavadee Aramvith, Teeranoot Chauksuvanit, and Yukari Shiota</i>	
Evaluation of A Flipped Classroom & Analysis of Students’ Learning Situation in a Computer-Programming Course 158	
<i>Yasuhiro Hayashi, Ken-ichi Fukamachi, and Hiroshi Komatsugawa</i>	
Time - A Multidimensional Concept .....	166
<i>Anneli Heimbürger</i>	
Grounded Multi-Level Computations .....	178
<i>Jaak Henno</i>	
An Explorative Cultural-Image Analyzer for Detection, Visualization, and Comparison of Historical-Color Trends 190	
<i>Yoshiko Itabashi, Shiori Sasaki, and Yasushi Kiyoki</i>	
Adaptive Systems for Multicultural Deployment .....	210
<i>Hannu Jaakkola and Bernhard Thalheim</i>	

Text Retrieval in SQL and No-SQL Environments .....	230
<i>Jevgenij Jakunshin, Antje Dusterhöft, and Christoph Eigenstetter</i>	
Abstraction Metaphors: A Unifying View of Modeling .....	235
<i>Roland Kaschek</i>	
Icon Recognition and Usability for Requirements Engineering .....	248
<i>Sukanya Khanom, Anneli Heimbürger, and Tommi Käykkäinen</i>	
Knowledge Support for Software Processes .....	261
<i>Michael Alexander Košinár, Jakub Štolfa, and Svatopluk Štolfa</i>	
A Metadata System for Quality Management .....	281
<i>Frank Kramer and Bernhard Thalheim</i>	
Designing Conceptual Database Models for Innovative Evaluation of Quality .....	300
<i>Elvira Immacolata Locuratolo</i>	
Relating Concept Theory to Computer Science .....	320
<i>Elvira Locuratolo and Jari Palomäki</i>	
Visualization of Ontologies on the Basis of Cognitive Frames for Knowledge Transmission .....	339
<i>Pavel Lomov and Maxim Shishaev</i>	
eLogika – The System for Teaching Logic .....	347
<i>Marek Menšík, Marie Duži, and Jakub Gerlich</i>	
A Data-driven Axes Creation Model for Correlation Measurement on Big Data Analytics .....	364
<i>Takafumi Nakanishi</i>	
An Adaptive Search Path Traverse for Large-scale Video Frame Retrieval .....	380
<i>Diep Thi-Ngoc Nguyen and Yasushi Kiyoki</i>	
Towards Finding Good Twitter Users to Follow Based on User Classification .....	399
<i>Tomoya Noro, Atsushi Mizuoka, and Takehiro Tokuda</i>	
An Axiomatic Approach to the Relational Concepts .....	407
<i>Jari Palomäki</i>	
Challenge in Urban Flood Mitigating System: Decision Support based on Cyber-Physical-Human Infrastructure 413	
<i>Dadet Pramadihanto, Wahyu Tjatur Sesulihatien, Soffi Patrisia, Saori Sasaki, and Yasushi Kiyoki</i>	
Design and Prototypical Implementation of an Integrated Graph-Based Conceptual Data Model .....	428
<i>Matthias Sedlmeier and Martin Gogolla</i>	
A Dengue Location-Contraction Risk Calculation Method for Analyzing Disease-Spread .....	448
<i>Wahjoe T Sesulihatien and Yasushi Kiyoki</i>	
Intelligent Software Support of the SCRUM Process .....	464
<i>Radoslav Štrba, Jakub Štolfa, Svatopluk Štolfa, and Michal Košinár</i>	

Generic Workflows - A Utility to Govern Disastrous Situations .....	473
<i>Marina Tropmann-Frick, Bernhard Thalheim, Diethard Leber, Gerald Czech, and Clemens Liehr</i>	
Mutual Resource Exchanging Model in Mobile Computing and its Application to Collective Intelligence 3D Movies	486
<i>Naofumi Yoshida</i>	
Development And Usage Of A Process Model Corpus .....	494
<i>Jürgen Walter, Tom Thaler, Peyman Ardalani, Peter Fettke, and Peter Loos</i>	
Context-Sensitive Framework for Visual Analytics in Energy Production from Biomass .....	508
<i>Pekka Warttainen, Anneli Heimbürger, and Tommi Kärkkäinen</i>	
Linguistic Rules for Automatic Summarization of Spoken Meetings .....	516
<i>Nils Weber, Christoph Eigenstetter, Antje Düsterhöft, and Markus Berg</i>	
Intercultural Collaboration in Virtual Environment .....	521
<i>Tatjana Weltzer, Hannu Jaakkola, Marko Hölbl, Marjan Družovec, and Anthony. E. Ward</i>	

## Preface

In the last three decades information modelling and knowledge bases have become essentially important subjects not only in academic communities related to information systems and computer science but also in the business area where information technology is applied.

The series of European – Japanese Conference on Information Modelling and Knowledge Bases (EJC) originally started as a co-operation initiative between Japan and Finland in 1982. The practical operations were then organised by professor Ohsuga in Japan and professors Hannu Kangassalo and Hannu Jaakkola in Finland (Nordic countries). Geographical scope has expanded to cover Europe and also other countries. Workshop characteristic - discussion, enough time for presentations and limited number of participants (50) / papers (30) - is typical for the conference.

Suggested topics include, but are not limited to:

1. **Conceptual modelling:** Modelling and specification languages; Domain-specific conceptual modelling; Concepts, concept theories and ontologies; Conceptual modelling of large and heterogeneous systems; Conceptual modelling of spatial, temporal and biological data; Methods for developing, validating and communicating conceptual models.
2. **Knowledge and information modelling and discovery:** Knowledge discovery, knowledge representation and knowledge management; Advanced data mining and analysis methods; Conceptions of knowledge and information; Modelling information requirements; Intelligent information systems; Information recognition and information modelling.
3. **Linguistic modelling:** Models of HCI; Information delivery to users; Intelligent informal querying; Linguistic foundation of information and knowledge; Fuzzy linguistic models; Philosophical and linguistic foundations of conceptual models.
4. **Cross-cultural communication and social computing:** Cross-cultural support systems; Integration, evolution and migration of systems; Collaborative societies; Multicultural web-based software systems; Intercultural collaboration and support systems; Social computing, behavioral modeling and prediction.
5. **Environmental modelling and engineering:** Environmental information systems (architecture); Spatial, temporal and observational information systems; Large-scale environmental systems; Collaborative knowledge base systems; Agent concepts and conceptualisation; Hazard prediction, prevention and steering systems.
6. **Multimedia data modelling and systems:** Modelling multimedia information and knowledge; Content-based multimedia data management; Content-based multimedia retrieval; Privacy and context enhancing technologies; Semantics and pragmatics of multimedia data; Metadata for multimedia information systems.

Overall we received 56 submissions. After careful evaluation, 16 papers have been selected as long paper, 17 papers as short papers, 5 papers as position papers, and 3 papers for presentation of perspective challenges.

We thank all colleagues for their support of this issue of the EJC conference, especially the program committee, the organising committee, and the programme coordination team.

The long and the short papers presented in the conference are revised after the conference and published in the Series of “Frontiers in Artificial Intelligence” by IOS Press (Amsterdam). The books “Information Modelling and Knowledge Bases” are edited by the Editing Committee of the conference.

We believe that the conference will be productive and fruitful in the advance of research and application of information modelling and knowledge bases.

Bernhard Thalheim  
Hannu Jaakkola  
Yasushi Kiyoki

## **Organisation**

### **Program Committee Co-Chairs and General Chair**

Programme committee has two operating co-chairs

- Yasushi Kiyoki, Keio University, Japan, and
- Bernhard Thalheim, Christian Albrechts University Kiel, Germany

and a permanent General Programme Chairman

- Hannu Kangassalo, University of Tampere, Finland.

### **Members of the Programme Committee**

The entire evaluation work has been performed by

- Bostjan Brumen
- Pierre-Jean Charrel
- Xing Chen
- Alfredo Cuzzocrea
- Marie Duží
- Anneli Heimbürger
- Jaak Henno
- Yoshihide Hosokawa
- Hannu Jaakkola
- Yasushi Kiyoki
- Sebastian Link
- Heinrich C. Mayr
- Tommi Mikkonen
- Jørgen Fischer Nilsson
- Tomoya Noro
- Jari Palomäki
- Matthias Riebisch
- Bernhard Rumpe
- Tetsuya Suzuki
- Bernhard Thalheim
- Peter Vojtas
- Yoshimichi Watanabe
- Naofumi Yoshida

Each reviewer had to review at least seven papers. Each paper got evaluated by three reviewer. Therefore the workload was very high for each of the PC members.

## **External Reviewers**

- Robert Eikermann
- Lars Hermerschmidt
- Katrin Holldobler
- Dennis Kirch
- Frank Kramer
- René Noack
- Alexander Roth
- Deni Raco
- Ove Sörensen
- Marina Tropmann-Frick

## **Organising Committee**

The general organising chair of this issue of the conference is

- Hannu Jaakkola, Tampere University of Technology (Pori), Finland

The organising committee

- Bernhard Thalheim, Christian-Albrechts University at Kiel, Germany
- Xing Chen, Kanagawa Institute of Technology, Japan
- Ulla Nevanranta (Publication), Tampere University of Technology (Pori), Finland
- Miklós Biró, Software Competence Center Hagenberg, Austria
- Klaus Pirklbauer, Software Competence Center Hagenberg, Austria
- Stefanie Jureit, Christian-Albrechts University at Kiel, Germany.

The organising committee was supported by the local organising team

- Steffen Gaede, Christian-Albrechts University at Kiel, Germany,
- Frank Kramer, Christian-Albrechts University at Kiel, Germany
- René Noack, Christian-Albrechts University at Kiel, Germany
- Ove Sörensen, Christian-Albrechts University at Kiel, Germany, and
- Marina Tropmann-Frick, Christian-Albrechts University at Kiel, Germany.

## **Programme Coordination Team**

The programme coordination has been managed by the two chairmen

- Naofumi Yoshida, Komazawa University, Japan and
- Anneli Heimbürger, University of Jyväskylä, Finland.



## The new organisation of the conference

A common observation for most conferences is that authors and contributors follow the pattern

*(1) come*

*(2) prepare for your talk*

*(3) don't listen to other talks*

*(4) be engaged with your next duties.*

This behaviour leads to a low benefit of the conference besides the publication and to a little benefit while submitting to and participating in this conference. Your paper will not get the attention it deserves.

## General schedule and organisation

This conference will however follow a different schedule and style. It is our aim to have a real interacting participants community at the conference. We thus extend the approach of the FoIKS ([www.foiks.org](http://www.foiks.org)), ADBIS 2010, iDB and other conferences with novel understanding of sessions, participation, discussion and interaction during and before and after the event, being a community by

- special sessions that bring researchers together and should result in co-authoring of future papers,
- implementing a different style of paper presentation so that everybody will understand all other contributions at the conference,
- special discussion forums before and after the paper presentation, and
- being already introduced to selected papers to be given at the conference before the conference.

It is our goal

- to enable active knowledge exchange within the modelling community,
- to stimulate cross-group research and publications,
- to produce a real scientific impact,
- to integrate young researchers into the scientific community,
- to overcome the shortcomings of traditional conferences, and
- to benefit from the experience of senior researchers from other groups.

We also target on

- making profit of the experience of senior researchers (based on iDB style),
- stimulating exchange with practioneers and potential users of new ideas,
- generating a set of good demonstration examples for future papers, and
- compilation of open problems in the area of modelling together with ideas how to tackle those problems.

Following the tradition of the EJC conferences long paper presentations have 30 minutes (25' for the talk), short paper presentations 20 minutes (16' for the talk), position paper presentations 15 minutes (12' for the talk), and challenges papers 10 minutes slots.

## The Appetiser session

This conference will have an appetiser session in which each participant with a talk can show what is the main message of the talk and why somebody should listen to it. This session allows everybody to be informed about all talks at the beginning of the conference. It allows each participant to capture the message, the achievements and the area of a conference paper.

## **The Collaboration Cluster Initiation session**

The second session will be a collaboration session in which a small group of participants explains to each other what is the essence of their talks, discusses what is of common interest and what should be the response to the given talk. The cluster groups will commonly exchange the papers of interest before the conference. Only after these two sessions we shall have the classical presentation program with one exception: groups are then posing questions, requests and thoughts first.

## **The Panel concept: Octavian Circle**

The panel discussions are going to be organised in an Octavian circle. It is similar to the way philosophers organise their discussions.

Octavian panel is a round table-table discussion with 8 (Octavian) people on the scene and a larger audience. Only those people can talk who are on the scene. The talk time is restricted to 3 minutes. Panellists may leave the scene at any time to join the audience, thus opening for a new panellist on the scene. There is a queue among participants in the auditory who want to contribute and to enter the panel. Anybody can leave or reserve a place in the queue. Questions from outside can be forwarded through the moderator. The moderator cannot leave the scene. The moderator's contribution is limited to 1 minute. The first two rounds of the panellists are pre-selected. This does not mean that this will be the order of contributing.

# Database Structure Modelling by Stereotypes, Pattern and Templates

Bader ALBDAIWI<sup>a,1</sup> and René NOACK<sup>b,2</sup> and Bernhard THALHEIM<sup>c,3</sup>

<sup>a</sup> *Dep. of Math. and Comp. Science, Kuwait University, Kuwait, Kuwait*

<sup>b</sup> *Christian-Albrechts-University Kiel, Dep. of Comp. Science, 24098 Kiel, Germany*

<sup>c</sup> *Christian-Albrechts-University Kiel, Dep. of Comp. Science, 24098 Kiel, Germany*

**Abstract.** Database research and practice has brought up a large body of knowledge and experience. This experience and knowledge is based on solutions for database structures that occur and reoccur in many applications in a similar form. We may distinguish two classes of such solutions: reference models that can be used as a blueprint for a fully fledged schema and stereotypes that are general solutions.

Pattern research considers structures at various levels of detail and is often limited to small schemata. Moreover, the abstraction level varies. We thus need a systematisation. This paper introduces stereotypes as general solutions to problems in a certain context, pattern as classes of refinements of such stereotypes, and templates as technology dependent solutions to problems in the given context.

We develop a general methodology and a number of techniques for stereotypes, pattern and templates. The paper considers structuring which is typically the starting point for development of database systems.

**Keywords.** pattern, template, modelling in the large, database structure, refinement and abstraction, extended ER models and schemata.

## 1. Structure Modelling in the Large

Database modelling does not start from scratch anymore. Typically, modellers reuse, extend or modify solutions that have already been developed or refine and adapt general solutions that have been developed by other modellers. Development of database structures can rely on the experiences of several decades of database realisation<sup>4</sup>. The body

<sup>1</sup>Corresponding Author: baderalbdaiwi@gmail.com <http://www.cs.ku.edu.kw/people/faculty/albdaiwi-b>

<sup>2</sup>noack@is.informatik.uni-kiel.de <http://www.is.informatik.uni-kiel.de/~noack>

<sup>3</sup>thalheim@is.informatik.uni-kiel.de <http://www.is.informatik.uni-kiel.de/~thalheim>

<sup>4</sup>Due to our involvement into the development and the service for the CASE workbenches (DB)<sup>2</sup> and ID<sup>2</sup> we have collected a large number of real life applications. Some of them have been really large or very large, i.e., consisting of more than 1.000 attribute, entity and relationship types. The largest schema in our database schema library contains of more than 19.000 entity and relationship types and more than 60.000 attribute types that need to be considered as different. Another large database schema is the SAP R/3 schema. It has been analysed in 1999 by a SAP group headed by the author during his sabbatical at SAP. At that time, the R/3 database used more than 16.500 relation types, more than 35.000 views and more than 150.000 functions. The number of attributes has been estimated by 40.000. Meanwhile, more than 21.000 relation types are used. The schema has a large number of redundant types which redundancy is only partially maintained. The SAP R/3 is a very typical example of a poorly documented system. Most of the design decisions are now forgotten. The high type redundancy is mainly caused by the incomplete knowledge on the schema that has been developed in different departments of SAP.

of knowledge developed so far and used in real practice is very large. It needs however a systematisation, categorisation and generalisation. There are very few publications (e.g. [21,24,34,44,43,45]) that provide such systematisation of the experience gained so far. The generalisation and the categorisation is however an open research field so far.

### 1.1. *Pattern as one of the Main Elements of Modelling in the Large*

Pattern are a means for systematisation, categorisation and generalisation. They generalise solutions that have been used over and over again before they are accepted as such. In general, a pattern [7] is a model [54] fragment that is profound and recurring. Pattern typically start with a frequently re-occurring problem, provide a notation for the solution and suggestions for its implementation. They provide some kind of generic structures that cover a generic problem.

There are many different notions for pattern [1,4,7,9,12,17,16,22,23,24,29,39,26,49], e.g.: (a) A pattern is a solution to a problem in context. (b) A pattern describes a particular recurring design problem that arises in specific design contexts, and presents a well-proven generic scheme for its solution. (c) A pattern is a template of interacting objects, one that may be used again and again by analogy. (d) A pattern is a proven solution to a common problem individually documented in a consistent format and usually as part of a larger collection. (e) A pattern is an idea that has been useful in one practical context and will probably be useful in others. (f) A pattern systematically names, motivates, and explains a general design that addresses a recurring design problem. It describes the problem, the solution, when to apply the solution, and its consequences.

Pattern and templates have already been developed for the conceptual modelling in the small [7] extending programming pattern widely discussed in the programming pattern research, e.g. [20,15]. Templates considered in [7,22,24,49] are simple structuring pattern such as products (tuples, records), finite sets, disjoint union, finite multi-sets, and finite lists. [7] considers archetypes that are special small core pattern or solutions for types such as account, actor, address, asset, contract, course, customer, document, event, flight, item, location, opportunity, part, position, product, role, transaction, and vendor.

Pattern research often distinguishes between *design* or *business-level pattern*, *conceptual-level pattern*, and *realisation* or *implementation pattern*. This distinction is follow the classical distinction into the business layer, the conceptual layer and the implementation layer of requirements, specification and coding. Often these three notions are used as synonyms or are mixed with each other. We may however generalise and systematise this distinction (1) by general solutions of a general problem within a context, (2) by implementation- and platform-independent conceptualisations of this general solution and (3) by platform-dependent templates. We this prefer to use three different notions: (1) *stereotypes*, (2) *pattern*, and (3) *templates*.

Pattern are a central element for *modelling in the large* which is based on *architectures* that allow to regard different view points such as modular structure of a model, context embedment of a model into associated systems, and ranges of involvement of components depending on business tasks, on *components* of an information system that reflect different tasks of the information system, on *generic solutions* for various parts of a schema in order to reuse similar solutions for similar problems in different parts of an application model, on *integration and collaboration facilities* for components, their interaction, and their common behaviour, on *distributed and embedded components* for

separation of a large system into services, sub-systems, and integrated sub-systems, on *methodologies* for building large schemata or models in a team which members have different roles, responsibilities and obligations during system development, and on *techniques for testing, verification and validation* of a large database or application system schema. Pattern may serve as the main abstract feature for architectures, components, and solutions. They provide a general solution and may thus be the kernel for generic solutions.

### 1.2. Pattern and their Advantages

Patterns, archetypes, pattern, templates and frameworks have been used in various ways. The pattern and framework literature is very rich. Starting with the seminal work of Alexander the 80ies and 90ies brought already deep insight, e.g., [25,1,4,9,12,17,16,23, 24,29,31,35,38,39,41,42,26]. A pattern [23,11] describes a commonly-recurring structure of communicating components that solve a general design problem in a particular context. For instance, presentation stereotypes in screenography [37] describe solutions for layout and playout problems tackled for web information systems development. These solutions are refined to pattern and pattern classes. Each pattern can then be refined to presentation grids which are conceptual models of the screen layout. The grid can then be mapped to templates which provide a logical model for the screen and interaction in web systems. Structuring of information systems is a classical topic and has got far more attention in research. We thus are able to present conceptions beyond those that have been developed for screenography. Advantages of pattern are:

- *Reusability*: Pattern enable a large scale of reuse and thus improve productivity, reduce development time and improve costs.
- *Communication*: Pattern support communication among developers and programmers.
- *Knowledge transfer*: Pattern capture design and development knowledge and design trade-offs. Pattern are prototypical fragments that distill the knowledge of experts.
- *Standardisation*: Pattern provide a standardised solution, e.g., by open standards.
- *Quality guarantee*: Pattern may be properly developed and thus may provide high-quality decisions. They improve uniformity and documentation. They reduce mistakes and rework.
- *Best practice*: Matured pattern provide the experience of large engineering communities and thus improve governance. They ease abstraction and thus modelling.
- *High-level views*: Pattern provide abstractions on realisations and thus close the gap between business and IT, minimise complexity and leverage IT skills.

[7] states that pattern should be small (consisting of less than 10 types). Modelling in the large is however based on larger components, e.g. the address pattern [49] consists of more than 20 entity and relationship types and more than 40 attributes beside the classical identification attributes.

### 1.3. Why Pattern are not Widely Used

Despite the good reasons discussed above we observe reasons why pattern are not as widely applied in practice and academic research as they should be:

- *Too small solutions*: Pattern research has already brought a deep understanding of basic structures such as trees and lists [7]<sup>5</sup>. These basic structures must however be combined and composed to larger structures. Otherwise they are deceptively too simple.  
Research problem (I): *Develop a set of main structures from which database structures may be composed.*
- *Pattern composition*: A database schema typically consists of a number of components which might be based on pattern. For this we need a schema algebra like the one in [32].  
Research problem (II): *Develop a pattern composition language that allows to construct schemata through expressions similar to inductive constructions in logic.*
- *Refinement*: Pattern need a coherent refinement of all elements in a pattern when we are going to integrate a pattern into a schema.  
Research problem (III): *Develop a systematic approach to coherent refinement of sets of structural pattern.*
- *Insufficient preciseness*: Some of the pattern are described at a very abstract level without sufficient support for definition of a precise pattern.  
Research problem (IV): *Pattern must be defined at the same time in narrative form, in visualised form, and in a formal way.*
- *Variation-proneness*: Pattern must be changeable without severe changes to others.  
Research problem (V): *Develop an approach to variation, evolution and migration of pattern-based structures similar to production lines.*
- *Structures without semantics*: Basic pattern such as those in [7] are representing the general structure without semantics of the types. Meaningful pattern are however those which carry at least some semantics and thus provide an intuitive understanding.  
Research problem (VI): *Develop pattern that are defined structurally, that use abstractions of semantics and that allow a pragmatic deployment.*
- *Direct path to realisation*: Pattern are often given at the high level or at most at the conceptual level. We need however a transformation approach for direct realisation. Code-level pattern for structures are known for reference models. These are however not adaptable to real applications.  
Research problem (VII): *Develop a suite of pattern that commonly support the business, conceptual and implementation levels.*
- *Complexity of structures generated by pattern*: General solutions are typically more complex than those developed in a monolithic form.  
Research problem (VIII): *Develop a tolerance level for deployment of pattern instead of monolithic specific solutions.*
- *Pattern adaptivity*: Pattern must be adapted to the specifics in the real application. The adaption must be flexible for the elements of the pattern.

---

<sup>5</sup>The six tree stereotypes are: hardcoded tree, simple tree, structured tree, overlapping trees, tree changing over time, and degenerated node and edge. They provide a basic structure similar to basic programming constructs. Trees are however the starting point for larger structures.

Research problem (IX): *Distinguish between mandatory, optional and extensible elements within a pattern.*

- *Anything at the same time:* Pattern proposals span from pattern at the business or design level to the coding level. They are thus tackling too many issues at the same time.

Research problem (X): *Develop an approach that allows to refine solutions at one abstraction level by several solutions at the next lower abstraction level.*

#### 1.4. Scope of this Paper

Patterns are not used as widely as they should be. Often they are too tiny [7]. Pattern composition, refinement, adaptation, and complexity are still open research issues. Partially, they are too imprecise. They must be variation-prone. Semantics must become a substantial element of pattern. We need a direct path to realisation. Moreover, overloaded patterns combine aspects of design, conceptualisation and realisation ('anything problem'). These problems result now in a good number of challenges. We concentrate in this paper on three of them for modelling of structures:

Research task (i): *Overcome the Anything problem (Problem X): Distinguish between a pattern at the business level ('general' pattern), a parameterised pattern at the generalised conceptual level and a pattern at the logical level ('specific' pattern).*

Research task (ii): *Provide real sufficiently large solutions (Problem I): Starting with basic pattern, pattern should be given for real life structures.*

Research task (iii): *Provide a procedure for stepwise refinement and integration (Problem III): Distinguish between general pattern, generalised conceptual pattern and logical pattern and provide a mapping facility for associating these different patterns.*

The first task is solved in this paper by a distinction into stereotypes, patterns and templates. The second task is solved for schema components following the subschema approach in [49]. Stepwise refinement is based on mapping of stereotypes to pattern classes and on mapping of patterns to template classes.

*Example (1). Throughout this paper we use a typical web information system: Digicult scout services. Scouts are collecting content for the Digicult system<sup>6</sup>. They interview witnesses and knowledgeable people, compare their narrations with content obtained so far, annotate the new content depending on the situation, and interact between each other. Furthermore, they search in museum archives and process notes made earlier by somebody. These scouts are typically people that are temporarily hired.*

## 2. Stereotypes as 'General' Pattern, Pattern as Conceptual Structures, and Templates as 'Specific' Pattern

The variety of pattern definitions is rather large. We integrate three of the main notions of the pattern and extend them by templates:

---

<sup>6</sup>The Digicult system is currently the museum portal of Schleswig-Holstein, Germany. The ISE@CAU team has been involved into the development of this system. It is currently extended for supporting demands of visitors and employees of museums and for information scouts who are collecting folklore knowledge.

- P1 [23,11] *A design pattern describes a commonly-recurring structure of communicating components that solve a general design problem in a particular context. It addresses a recurring design problem that arises in specific design situations, and presents a solution to it. Pattern document existing, well-proven design experience. Patterns identify and specify abstractions that are above the level of a single class and instances, or of components. Patterns provide a common vocabulary and understanding for the design principles. Patterns are a means of documenting software architectures. Pattern support the construction of software with defined properties. Patterns help to build complex and heterogeneous software architectures. Patterns help you to manage software complexity.*
- P2 [10,47]: *A pattern describes a particular recurring design problem that arises in specific design contexts, and presents a well-proven generic scheme for its solution. The solution scheme is specified by describing its constituent components, their responsibility and relationships, and the ways in which they collaborate.*
- P3 [2,3]<sup>7</sup>: *Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution. It is a relationship between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain spatial configuration which allows these forces to resolve themselves. It is an instruction, which shows how this spatial configuration can be used, over and over again, to resolve the given system of forces, wherever the context makes it relevant. A stereotype is, in short, at the same time a thing, which happens in the world, and the rule which tells us how to create that thing, and when we must create it. It is both a process and a thing: both a description of a thing that is alive, and a description of the process which will generate that thing.*

Our separation of concern into pattern at the business level ('general' pattern), pattern at the generalised conceptual level and pattern at the logical level ('specific' pattern) allows to develop a general approach to pattern and to develop a technology for each kind of such pattern. It supports treatment of pattern in a more sophisticated way.

Assumption (a): *Pattern at the business level handle in a general way the solution for a problem.*

Postulate (A): *A stereotype is a general pattern at the business level. It can be mapped to a variety of (conceptual) pattern.*

*A structure stereotype describes data structuring solutions to problems within a certain context. These solutions can be refined and used in systems.*

Assumption (b): *Pattern with parameters are those at the conceptual level.*

Postulate (B): *A pattern is an implementation- and platform-independent general conceptual solution that can be mapped to different realisations.*

*A structure pattern (or structure archetype) is based on paradigms and principles in certain modelling environment and specialises the stereotype by adding control and support.*

Assumption (c): *Pattern at the logical level provide a solution with consideration of a given platform.*

Postulate (C): *A template is a platform-dependent (and implementation-oriented) pat-*

---

<sup>7</sup>The architect Christopher Alexander introduced the notion of the pattern in 1977 and extended this notion to software engineering, e.g. in his invited speech at OOPSLA'96. The pattern concept is widely used in architecture since the 70ies, especially for rural development and architectures of buildings.

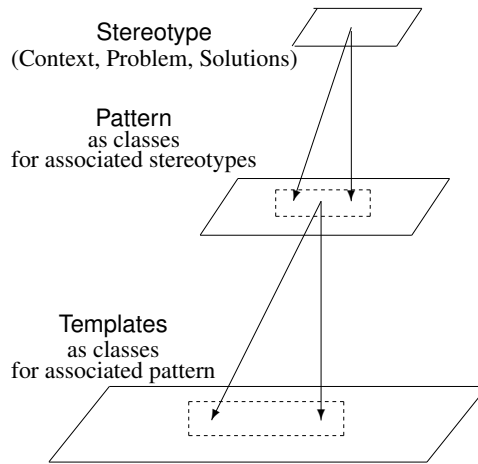


tern.

A *structure template* may embed pattern into the technology and provides a parametric ready to use solution. Some reference models or schemata are typical examples of templates.

Assumption (d): *Stereotypes are refined to pattern within a pattern class. Pattern are refined to templates within a template class.*

Conclusion (A): *A stereotype can be associated to a class of pattern. A pattern can be associated to a class of templates.*



Stereotypes should provide a solution to a problem. This problem carries however also some context. Stereotypes thus provide a set of solutions for a given problem in a certain context. A class of pattern is associated with a given recipe. These classes might overlap for different recipes. Templates are refinements of pattern and thus can also be associated to pattern. A class of templates is associated to a given pattern. These classes can also overlap. Therefore, templates associated with a stereotype are the union of classes for all pattern that are associated to a stereotype.

Refinement is definable through the government and binding approach<sup>8</sup> and functionalisation. In this case, refinement is based on context extension, instantiation of parameters and specialisation of components of the structural expression which are the main dimensions of function derivation starting with a generic function.

Functionalisation has been introduced by A. Bienemann [5] for characterisation of general collections of functions with a similar meaning and behaviour. These 'generic' functions can be refined to functions within their collection. Consider types  $t_1, \dots, t_k, t'_1, \dots, t'_n$  for  $k, n \in \mathbb{N}$ . Consider  $Dom := Col^{t_1} \times \dots \times Col^{t_k}$  and  $Rng := Col^{t'_1} \times \dots \times Col^{t'_n}$  to be sets of tuples of collections of objects of the corresponding types, for  $k, n \in \mathbb{N}$ . Consider a function  $f : Dom \rightarrow Rng$  mapping a tuple

<sup>8</sup>The government and binding theory [13,14,46] was an international effort that aimed at a universal grammar for most verb-based and state-oriented natural languages. It was based on the observation that humans typically reason based on their concept space. This reasoning is almost independent on a given language. Research on cognition is nowadays assuming that most concepts are constructed as associative maps. Therefore, concepts can be defined along the line that has been proposed by G. L. Murphy [36]. Then an utterance can be build in dependence on the specific features and capabilities of a specific language.

Therefore, an utterance is constructed in a three step process (see Figure 3):

- (a) construction of a cognitive structure (D-structure);
- (b) mapping of the D-structure by a number of rules (called  $\alpha$ -rules) to a specific structure (S-structure)
- (c) mapping of this S-structure to the specific utterance based on another language specific set of rules.

The universal grammar programme finally failed - mainly however due to the mathematical difficulties of such a general approach. The construction process for utterance is however based on cognitive psychology. The D-structure can be language independent and thus allows to talk at the same time in several different languages without revision of the sense and meaning of the contribution.

of collections of types  $t_1, \dots, t_k$  into a tuple of collections of types  $t'_1, \dots, t'_n$ . Consider two formulae  $\phi, \psi$  defined over the domain and the co-domain of  $f$ , respectively. The quintuple  $(Dom, \phi, f, \psi, Rng)$  is called a *function application* with precondition  $\phi$  and postcondition  $\psi$ . Function applications can be defined recursively by means of operators of the function algebra of choice, i.e.,  $(Dom, \phi, \theta(F_1, \dots, F_m), \psi, Rng)$  is a function application, if  $\theta$  is a corresponding operator of the function algebra, and  $F_1, \dots, F_m$  are function applications.

A *functionalisation* is a quadruple  $(\mathcal{S}, \mathcal{F}, \Sigma, s_0)$  with

1. a specification of *structuring* denoted by  $\mathcal{S} = (S, V)$  and consisting of a *database schema*  $S = (T^S, \Sigma^S)$  with  $T^S$  being a set of types according to the type system as stated above, and  $\Sigma^S$  being a set of static integrity constraints, and a set of views  $V$  upon schema  $S$  defining collections in domains, and co-domains of function applications,
2. a function application  $\mathcal{F}$ ,
3. a set of dynamic integrity constraints  $\Sigma$  on  $\mathcal{S}$ ,
4. a distinguished initial state  $s_0$ .

Stereotypes, pattern and templates are based on languages. Since structuring can be based on stepwise construction we may use basic solutions and must provide a number of constructors for incremental construction of database schemata [32,33]. Since problems have their own scope and granularity we may not follow classical database engineering. Stereotypes, pattern and templates already define schemata as the basis. So, we cannot decompose solutions in general.

Stereotypes, pattern, and templates are models [18,50,51,53,52]. Their level of detail varies. Models can be refined and composed. Reference models are typical neither stereotypes nor pattern nor templates. They are specific within an application domain. They are thus application dependent. They can be used as starting point for the development of a schema for a specific database application. They tend to become rather large since their richness allows to consider many facets for an application area. They are less abstract than stereotypes or pattern. Templates may however be enriched to become reference models.

Stereotypes, pattern and templates are evaluated. We follow [7] and require that these need to be proven solutions that has stood the test of time. Additionally we might develop a number of quality characteristics [27,28].

We cover the three prominent pattern approaches P1,P2,P3 used in software engineering. The first approach refers to pattern. The second and third approaches refer to stereotypes. None of them considers templates.

### 3. Stereotypes as (Problem,Context,Solution)-Triples

Modern software engineering [8] separates the products of a software development process into *abstract problem models* that describe problems in an abstract form, *abstract solution models* that provide solutions which are implementable and system-independent, and *executable solutions* that are refinements of the the abstract solution in dependence on the systems used, on their architecture, on the specific languages, etc. The first model covers the classical analysis phase, the second the design phase, and the third one the implementation phase. This refinement process from abstract problem to concrete solution

is typically bound by the *context* which is often taken for granted. We thus use the triple

( *problem, context, solution* ) .

### 3.1. Problems

Problems are given in a natural language by four components:

(a) The *state space* consists of the collection of all those states that are reachable from the *initial state*. Some of the states are considered to be desirable, i.e. are *goal states*. States can be modelled through languages such as ER. (b) *Actions* allow to move from one state to another state under certain conditions. (c) The *goal test* determines whether a given state or state set satisfies the goals. (d) The *problem solution controller* evaluates the actions undertaken by the user. Some solutions may be preferred over other, e.g. have less costs, or are optimal according to some optimality criterion.

*Example (2).* One of the problems to be solved in the Digicult project is to find a way to store and to collect any document that has been used to characterise exhibition items. Other similar problems are the support for in-situ interviews, the seamless integration of scout collections, and the enquiry of hidden information and knowledge. The initial state is characterised by a large and not entirely specified set of document kinds. The goal state is a database schema that allows to integrate any information from these documents. The goal test allows to check whether all known document kinds can be reflected by the database schema and whether any unknown document kinds can also be supported. The controller may be used to check redundancy of data and structures.

### 3.2. Context

According to [56] we distinguish between the *intext* and the *context* of things which are represented by objects. Intext reflects the internal structuring, associations among types and sub-schemata, the storage structuring and the representation options. Context reflects general characterisations, time, application domain, categorisation, utilisation, and general descriptions such as quality.

*Example (3).* The context space must reflect the wide variety of documents used in museums, e.g., notes, record cards, register entries. Documents might be related to the personal, to space and time, to evolution history, to deployment, etc. Documents are superimposed by comments. Since we are in the museums domain we might use a variety of categorisation dictionaries, ontologies and specific namespaces.

### 3.3. Solutions

Solutions may be enhanced by illustration of the solution, examples, and behaviour and structure of the solution. We use the classical rhetoric frame introduced by Hermagoras of Temnos<sup>9,10</sup> and characterise a *solution* by answering the questions: Who, what, when, where, why, in what way, by what means.

<sup>9</sup>Quis, quid, quando, ubi, cur, quem ad modum, quibus adminiculis

His work is almost lost. It is however reflected in the work of his followers, e.g., Cicero. Before Aristoteles already used topics, foci, viewpoints, styles, and scopes for rhetoric functions.

<sup>10</sup>Computer engineering considers J. A. Zachman [57] as the inventor of this framework and of foci of models despite the well-known frame used since two thousands of years.

The solution is given by an informal characterisation, the state space, the required and commanded behaviour, and interactions. An additional characterisation describes applicability of the solution, quality characteristics, and consequences imposed by the solution. Result elements describe the results obtained by the solution. Additionally, solutions might be related to other similar solutions.

The solution space can be restricted by *principles* which should be observed. Principles can be classified into soft principles and strict principles.

*Example (4).* One solution for the document storage problem in the Digicult project can be the definition of a collector that allows to electronically store any document of known kind and that allows to electronically collect all other documents as well. One more specific solution could be a storage of any document and a collector of those documents which kind is known so far.

```
stereotype COLLECTOR (S, IV, C, VV, VL, M, S, R, F)
  uses view-for-integration V into database for schema S
  uses data-receival-controller C
  uses data-verification-view V for data-confirmation-log VL
  uses monitor-for-display M
  display status S
  react R on failure F
```

A typical activity is the play of a role which can be specified by the rhetoric frame. It can be enhanced by answering : “wherefore” (purpose), “whereof” (origin), “wherewith” (carrier, e.g., language), and “worthiness” ((surplus) value). Activities (“how”) describe the way how the work is performed and the practises for the work. Work products (“what”) are the result of the activity. Roles (“who”) describe obligations and permissions, the involvement of parties into the activity. Aspects (“where”) are used for separation of concern in the activity. Resources (“on which basis”) are the basis for the activity.

### 3.4. Stereotypes

Definition (1): A *stereotype*  $(\mathcal{W}, \rho, \{S\})$  consists of a class (or collection or set)  $\{S\}$  of solutions to a problem  $p$  within some context  $\mathcal{W}$  :

- A problem  $\rho$  is specified by a frame  $\mathcal{F} = (S, A, G, C)$  that consists of a non-empty set  $S$  of states, a set of actions  $A$  for transformation of a state to another state, a goal test  $G$ , and a controller  $C$  for application of actions to states.
- A solution  $S$  consists of an expression  $E$  defined on  $A$  such that the current state  $S$  can be faithfully transferred to a state  $S'$  by  $E$  for which the goal test is valid, i.e.  $S' = E(S), G(S') = \text{true}$ .

A transformation is faithful if the controller allows to apply each action in  $E$  in the order given by  $E$ .

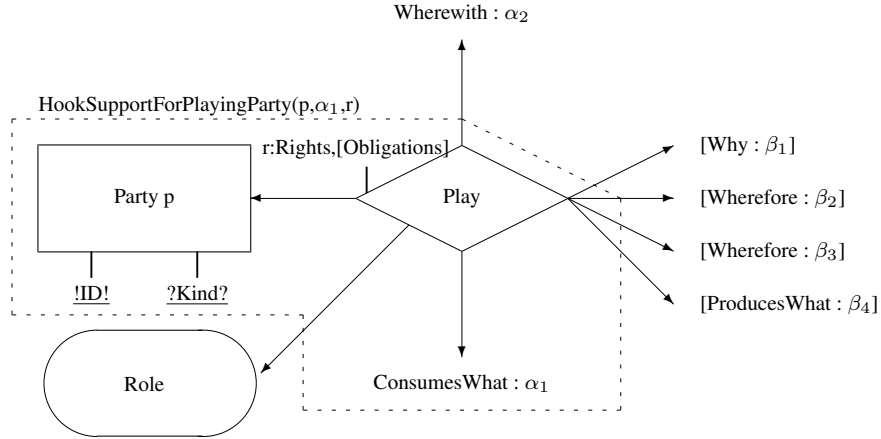
## 4. Structure Pattern

Pattern can be represented by schemata in an enhanced higher-order entity-relationship model [48]:

*Optional elements* (represented by square brackets) may be removed in the specialisation process. *Obligations* (depicted by exclamation marks around the type name) denote

questions that should be resolved during specialisation to a template or to a conceptual schema. *Hocks* (denoted by dotted polygons within the schema with parameters) are views on a schema. They can be anchored by other schemata. *Anchors* and hocks can be parameterised. *Links* (represented by labelled parameters) are either parametric components of a type or explicit links to anchors. We could also distinguish different kinds of links. *Behavioural structures* (represented by rectangles with rounded corners) allow to track a story. They are typically reflected by own schemata and thus folded into the schema during the refinement process. *Open issues* (depicted by question marks around the type name) are markups for later solution and integration of other pattern.

*Example (5).* Let us consider the general pattern in Figure 1 which is used for storing certain activities during collection as a component of a collector. A role is a part or character played by an organisation or a person. Roles can be categorised. Roles are typically associated with rights, obligations, permissions and forbidden actions. The rela-



**Figure 1.** Pattern: A party plays a role.

relationship type `play` can be refined to  $n$ -ary relationship types for  $n \in \{2, 3, 4, 5, 6, 7, 8\}$ . The behavioural structure `role` forms its own schema. It can be refined to an entity type `RoleKind`. It could however also be represented by a schema that reflects all elements of roles.

Additionally we add controls  $\mathcal{C}$  and supports  $\mathcal{U}$ . The control specification is divided into a restriction of the scope of control, the description of the quality control tasks, participants or controllers, the execution context, and restrictions applied to control. Control specification can be formally defined by dynamic integrity constraints. Main support functions are `insert`, `delete`, `update`, and `read` that can be applied to each element of a pattern and to the pattern. Database technology uses policies for referential integrity control such as `cascade`, `restrict`, `no action`, `set null`, and `set default`. Since we allow several default values the last policy uses a parameter with a default value.

**Definition (2):** The *pattern declaration* consists of a name of the pattern  $N$ , the schema of the pattern  $S$ , deployment conditions  $\Psi$ , integrity constraints  $\Sigma$ , the parameters  $p_i$  of the pattern with their pre- and post-conditions  $\gamma_i, \delta_i$ , the controls

$\mathcal{C}$  for all functions that can be applied to the pattern, and the supports  $\mathcal{U}$  for all functions that can be applied to the pattern, i.e.

$$\mathcal{P} = (N, S, \Psi, \Sigma, \mathfrak{F}, \mathcal{C}, \mathcal{U}) \quad .$$

*Example (6). The pattern is now enhanced by adding control policies. The policies given in the table below can be called moderately liberal. The support for this pattern may be a lazy one, i.e. there are no specific support mechanisms for the elements of the pattern itself. Components of the pattern inherit the support of the given component.*

	insert	delete	update	read	support
<i>HookSupportFor</i>					
<i>PlayingParty</i> ( $p, \alpha_1, r$ )	cascade	set default	restrict	full	lazy
<i>Party</i>	no action	restrict	cascade	full	inherit
<i>Rights, [Obligations]</i>	cascade	cascade	cascade	full	lazy
$\alpha_1$	cascade	set default	restrict	view( $v_1$ )	inherit
$\alpha_2$	restrict	restrict	restrict	view( $v_2$ )	inherit
$\beta_1$	restrict	restrict	restrict	full	inherit
...	...	...	...	...	...

Pattern can be composed to become larger pattern. The composition of pattern follows the approach developed in [32,33]. Given a number of pattern  $\mathcal{G}_1, \dots, \mathcal{G}_k$  with parameters  $p_{1,1}, \dots, p_{1,m_1}, p_{2,1}, \dots, p_{k,m_k}$  and their pre- and post-conditions. A parameter  $p_{i,j}$  with conditions  $\gamma_{i,j}, \delta_{i,j}$ , can be assigned to a pattern  $\mathcal{G}_n = (N_n, S_n, \Sigma_n, \mathfrak{F}_n, \mathcal{C}_n, \mathcal{U}_n)$  only if  $\Sigma_n \models \gamma_{i,j}$  and if  $\delta_{i,j}$ , is satisfied after assignment. The deployment condition of the resulting pattern is  $\Sigma_n \cup \Sigma_i \cup \{\delta_{i,j}\}$ .

Pattern are related to each other. We distinguish between primary associations and secondary associations. Typical primary associations are *uses*, *refines*, and *conflicts*. Secondary associations are *variant*, *variation*, *similar*, *combines*, *requires*, *tiles* (uses itself recursively), and *alternative\_of*.

## 5. Structure Templates

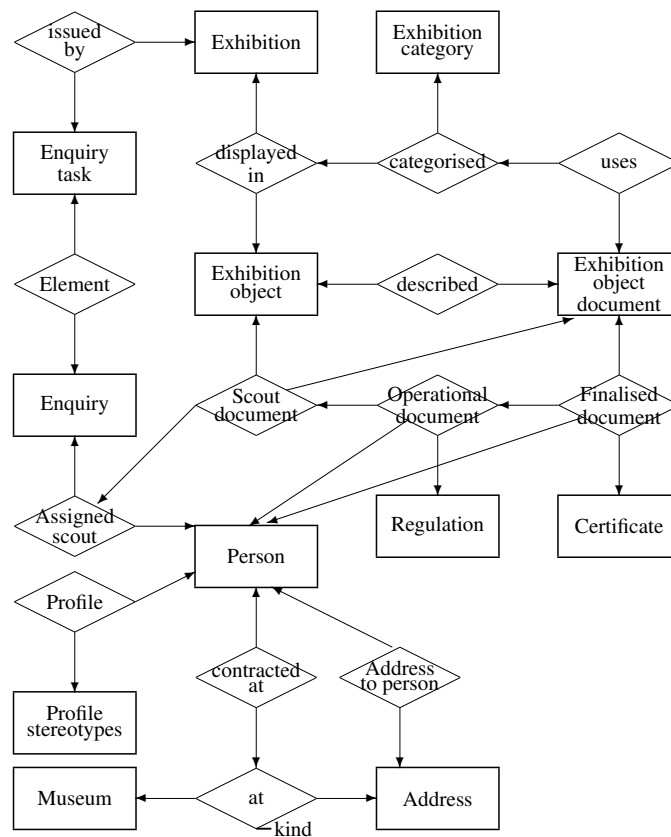
Templates are refinements of pattern and use a general description of technology envisioned for realisation. We may, for instance, base a template on object-relational technology with ER schemata, on XML technology with XSchema, and on form type technology for big data. Form types are named tree structures whose nodes are component types. Each component of a form type has a non-empty set of attributes and keys, a possibly empty set of constraints, and a set of database operations. Templates are named ( $N$ ) and use a description of their structures  $\mathcal{S}$ . For instance, the structure specification may consist of a schema and queries defining views on top of the schema.

*Example (7). Figure 2 displays the schema of the object-relational template for the scout collection*<sup>11</sup>. It uses the pattern in Figure 1 three times for scout document, for operational document, and for finalised document. For instance, the first two applications are

<sup>11</sup>We use the extended entity-relationship model (HERM) developed in [48] for the representation. It generalises the classical entity-relationship model by adding constructs for richer structures such as complex nested attributes, relationship types of higher-order  $i$  which may have relationship types of order  $i - 1, i - 2, \dots, 1$  as their components, and cluster types that allow disjoint union of types.

given by the following refinements of the pattern parameters:

Pattern parameters	Play for scout document	Play for operational document
ConsumesWhat	Exhibition object documents	Scout document
Wherewith	Exhibition object	Exhibition object (inherited)
Party	Scout	Museum custodian
Role	Collector	Document assembler
Why	Enquiry task	Enquiry task (inherited)
Wherefore	Reason: Missing exhibition object document	Trigger: Scout document
Wherefore	$\lambda$	BasedOn: Regulation
ProducesWhat	Scout document	Operational document



**Figure 2.** Object-relational template schema for scout data on museum notes (without attributes, roles and constraints)

The schema refines the role behavioural structure. It uses a pattern for collections. This pattern is refined to an exhibition object subschema. The role behavioural structure embeds a person characterisation subschema. Furthermore, a task subschema is used.

Templates are enhanced by *realisation styles and tactics S&T*. They refine controls and supports of pattern. For instance, ER schemata are based on a rigid class/type separation and support user viewpoints through views. We use thus a local-as-view style for views. Constraint tactics declare how eager (e.g., cascade, restrict) constraints are enforced, at what time frame (e.g., immediate) and with which kind of enforcement (e.g.,

check before operation). Object-relational technology also supports tactics such as configuration (central, local, backup), recovery policy, storage type, redundancy, and ownership style. Tactics may also include statements such as whether controlled redundancy is supported, which kind of stored procedures is preferred, and which directives for constraint enforcement are given. Tactics may be enhanced by abstract description of requested tuning techniques.

The technology embedding includes configuration parameters  $\mathcal{C}$ . They declare coding (e.g., character sets), services (e.g., directory services, wallet services, listeners), policies (e.g., memory policy, user policy), and handlers (e.g., backup schedule). Configurations may be enhanced by hints  $\mathcal{H}$ . Typical hints in DBMS are hints for querying and modification.

The co-design approach [48] integrates structure specification with functionality specification. This functionality may be defined in a generic form similar to classical generic operations such as `insert`, `update`, `delete`. Typical generic operations  $\mathcal{O}$  are `retrieve`, `modify`, `combine`, `display`, `select_from_collection`. Modern systems also must provide information on the quality  $\mathcal{Q}$ , the capability and reliability of data in the system.

**Definition (3):** A *template*  $\mathcal{T} = (N, \mathcal{S}, S\&T, \mathcal{C}, \mathcal{H}, \mathcal{O}, \mathcal{Q})$  is given by a name  $N$ , a structure or schema  $\mathcal{S}$ , realisation styles and tactics  $S\&T$ , configuration parameters  $\mathcal{C}$ , hints  $\mathcal{H}$ , generic operations  $\mathcal{O}$ , and quality characteristics  $\mathcal{Q}$ .

Templates are constructed from pattern by application of refinement operations such as `refine` (the pattern until all obligations are fulfilled and parameters are assigned), `forget` optional components, `use` a pattern by completing the pattern parameters, `extend` parameters and behavioural structures and adding hints, `instantiate` parameters and values, `assign` data types to atomic attributes in a pattern, `resolve` obligations (!...!) and open issues (?...?), `inject` schemata for behavioural structures, `controller` for quality assurance such as consistency support, and `supporter` for efficient management of the objects.

*Example (8).* Figure 2 uses a schema of the object-relational template for the scout collection. We can also use the schemaless big data template that consists of a set of triples:

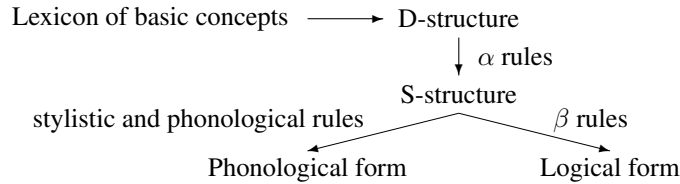
(context, parameter, value) .

These triples can be formally based on the notion of the tuple space.

A tuple space  $(\mathcal{D}, \mathcal{C}, (\mathcal{R}, \rho, \theta, \Psi), \mathcal{G}, \mathcal{W})$  uses triples (things, have\_concepts, with\_validity) and is given

- by things  $\mathcal{D}$  under consideration
- by concepts  $\mathcal{C}$ , properties, etc., described in a language  $\mathcal{L}$ , and
- by a “validity” relationship  $\mathcal{R} \subseteq \mathcal{D} \times \mathcal{C}$  with
  - restrictions  $\rho$  to its applicability,
  - a modality  $\theta$  or rigidity of the relationship, and
  - a confidence  $\Psi$  in the relationship
- that is agreed upon within a group  $\mathcal{G}$  within a culture  $\mathcal{C}$ , and
- that is valid in a certain world  $\mathcal{W}$ .





**Figure 3.** The levels of representation and rules used in government and binding

The XML template is based on tactics [30]. The extended entity-relationship model uses, for instance, a strict separation of structural model elements into attribute, entity, relationship, and cluster types. This separation is a variant of the ‘Salami slice’ form of XML documents that require that each document represents an object and that uses an ensemble of XML documents which are interrelated by references. XML models may be based on the so-called ‘Venetian blind’ or ‘Russian doll’ representation. The later specific form requires that documents are closed in the sense that all essential references to other documents are replaced by those documents within the given document.

## 6. Systematic Treatment of Stereotypes, Pattern and Templates

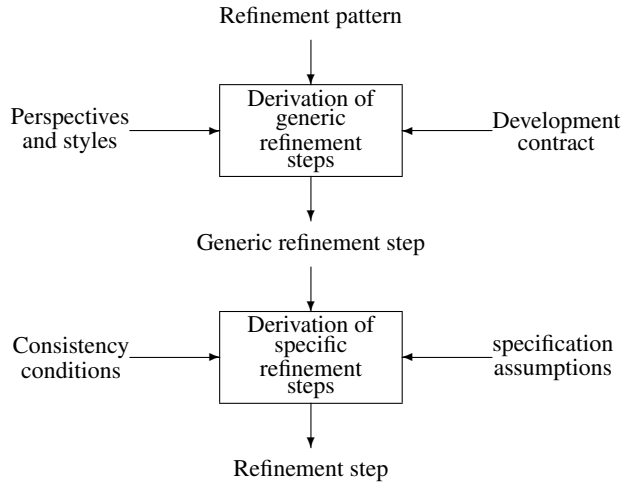
Refinement of structuring is based on the generalisation of Government and Binding developed in [5,6] and on the refinement of schemata [55] which generalises refinement developed for abstract state machines. This style of refinement is already applied to screenography in [37].

A typical engineering approach to development of work products such as programs or specifications is based on a general methodology, operations for specification evolution, and a specification of restrictions to the modelling itself. Each evolution step must either be correct according to some correctness criterion or must lead to obligations that can be used for later correction of the specification. The correctness of a refinement step is defined in terms of two given system together with the equivalence relations. Already in [40] it has observed that refinement steps can be governed by contracts. We may consider a number of governments [6] in the sense of [13]. However we should take into account the choices for style and perspectives.

Refinement may be based on the linguistic theory of Governance and Binding (GB), which consists of a two-step specialisation of ideas or raw utterances [13]. GB assumes that a universal grammar can be broken into two parts: levels of representation and a system of constraints. It assumes a derivational model and four different levels of representation (see Figure 3). The lexicon lists the atomic units of the syntax called basic concepts. Lexical items are combined together to a D-structure, which might be a forest of concept fields [19]. D-structures are mapped into S-structures that reflect the syntactic surface order of the sentence. Examples of S-structures are query and answer forms [48], which are used for representing the general structure and functionality of a given query. S-structures are factored into phonological forms and logical forms. The former ones may be understood to be specific representations of the sentence; the latter ones combine the interface with semantics.

Given a refinement pattern, perspectives, styles and contract, we may derive generic refinement steps such as data refinement, purely incremental refinement, submachine

refinement, and (m,n) refinement. The generic refinement is adapted to the assumptions made for the given application and to consistency conditions. Typically such consistency are binding conditions of rules to state and vocabulary through the scope of rules. The general approach we envision is depicted in Figure 4.



**Figure 4.** The Derivation of Correct Refinement Steps

Figure 5 sketches thus the general process of stereotypes selection, pattern derivation and evaluation, and template derivation and evaluation. We target at a development of libraries. The sets of stereotypes, pattern and templates are not going to be complete. They may only be sufficient for certain applications. We may associate stereotypes and pattern from one side an pattern and templates from the other side. Each stereotype can be refined or specialised by some program called transformer to pattern which are refinements in the sense of [55]. In a similar form we may associate pattern and templates. Both associations are not exclusive. One pattern may for instance refine several stereotypes. This association results in classes of pattern  $\mathcal{P} \in class(\mathcal{P})$  for a specific stereotype  $\mathcal{R}$ , and to classes of templates  $\mathcal{T} \in class(\mathcal{T})$  for a specific pattern  $\mathcal{P}$ . Classes of pattern may intersect; the same is true for classes of templates.

## 7. Conclusion

We presented a general approach to schema pattern. We distinguish between stereotypes as pattern at the business level, pattern at the generalised conceptual level, and templates at the logical level. This distinction allows to relate stereotypes to a class of pattern and to relate pattern to a class of templates. The composition of pattern and templates is based on the schema algebra [32,33]. This approach is related to model-driven development and model-driven architectures. We do not consider the mapping to logical structures since this mapping depends on many realisation decisions. Stereotypes are however not models in the general sense. We limit the consideration in this paper to modelling of structures and defer modelling of semantics to a later paper.

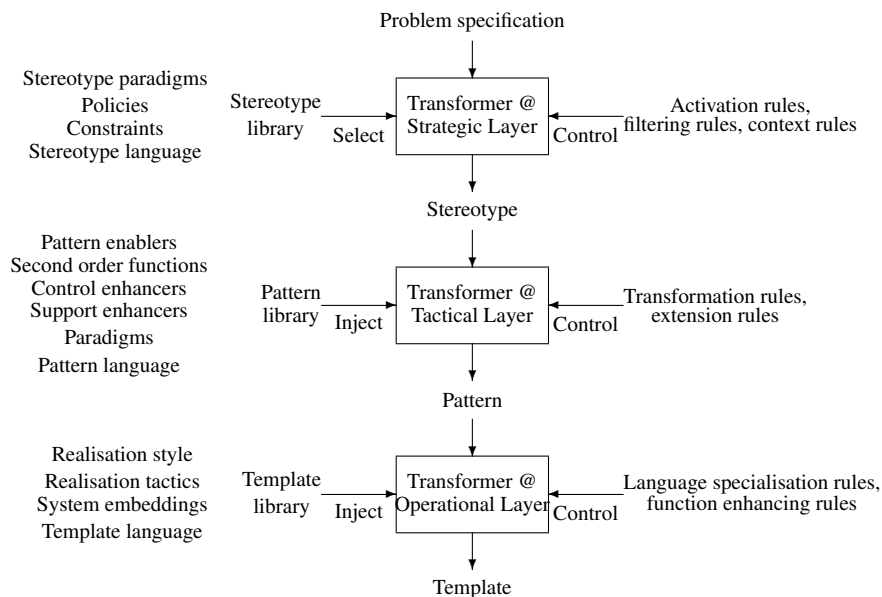


Figure 5. Stereotype, pattern and template generation

Extending the approach in [49] for production management systems, we have used this approach for redevelopment of large database structures observed in web service portals, health care applications, and financial applications. We observed pattern for structures as well as pattern for construction in these applications. The redevelopment of such schemata has been carried out in Bachelor theses of students, i.e., within a very limited time frame. The schemata span from schemata with few hundreds of attribute, entity, relationship and cluster types to such with more than 1000 types.

There are many open research issues, for instance, the following ones:

**Integration of constraints into the schemata.** A solution for structures with integrity constraints can be based on dynamic sets of integrity constraints [48].

**Systematic development of schemata with initial architectures of a schema.** A systematic methodology for such general pattern-driven development is necessary for modelling in the large.

**Stereotypes, pattern and templates for static semantics of database schemata.** Controls and supports discussed above highlight the way how pattern for sets of integrity constraints can be developed. Such pattern will ease handling of integrity.

## References

- [1] P. Aiken. *Data Reverse Engineering: Slaying the Legacy Dragon*. McGraw-Hill, 1995.
- [2] Christopher Alexander. *The Timeless Way of Building*. Oxford Press, 1979.
- [3] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1997.
- [4] K. Beck and R. Johnson. Patterns generate architectures. In *Proc. ECOOP'94, Bologna, Italy, 1994*. <ftp://st.cs.uiuc.edu/pub/patterns/papers/patterns-generate-archs.ps>.
- [5] A. Bienemann. *A generative approach to functionality of interactive information systems*. PhD thesis, CAU Kiel, Dept. of Computer Science, 2008.

- [6] A. Bienemann, K.-D. Schewe, and B. Thalheim. Towards a theory of genericity based on government and binding. In *Proc. ER'06, LNCS 4215*, pages 311–324. Springer, 2006.
- [7] M. Blaha. *Pattern of Data Modelling*. CRC Press, Boca Raton, 2010.
- [8] B. Boehm. A view of 20th and 21st century software engineering. In *Proc. ICSE'06*, pages 12–29, ACM Press, 2006.
- [9] J. Bosch, P. Molin, M. Mattsson, and P. O. Bengtsson. Object-oriented frameworks: Problems & experiences. <http://www.ipd.hk-r.se/michaelm/papers/ex-frame.ps>, 1997.
- [10] F. Buschmann, K. Henney, and D.C. Schmidt. *Pattern-oriented software architecture - A pattern language for distributed computing*, volume 4. Wiley, Chichester, 2007.
- [11] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-oriented software architecture - A system of patterns*, volume 1. Wiley, Chichester, 1996.
- [12] R. H. Campbell, N. Islam, and P. Madany. Choices, frameworks and refinement. *Computing Systems*, 5(3), 1992. <ftp://choices.cs.uiuc.edu/Papers/Journal/Compsys.ps>.
- [13] N. Chomsky. *Some concepts and consequences of the theory of government and binding*. MIT Press, 1982.
- [14] N. Chomsky. *The minimalist program*. MIT Press, Cambridge, 1995.
- [15] J. Coplien and D. Schmidt. *Pattern Languages of Program Design*. Addison-Wesley, 2005.
- [16] J. O. Coplien and D. C. Schmidt, editors. *Pattern languages for program design*. Addison-Wesley, Reading, 1995.
- [17] S. Cotter and M. Potel. *Inside Taligent technology*. Addison-Wesley, 1995.
- [18] A. Dahanayake and B. Thalheim. Enriching conceptual modelling practices through design science. In *BMMDS/EMMSAD*, volume 81 of *Lecture Notes in Business Information Processing*, pages 497–510. Springer, 2011.
- [19] A. Düsterhöft and B. Thalheim. Integrating retrieval functionality in websites based on storyboard design and word fields. volume 2553 of *LNCS*, pages 52–63. Springer, 2002.
- [20] F. Buschmann et al., editor. *Pattern Oriented Software Architecture: A System of Patterns*, volume 1. Wiley, 2001.
- [21] M. Fowler. *Refactoring*. Addison-Wesley, Boston, 2005.
- [22] M. Fowler. *Anlysemuster*. Addison-Wesley, 1999, Bonn.
- [23] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: Elements of reusable software architecture*. Addison-Wesley, 1995.
- [24] D. C. Hay. *Data model pattern: Conventions of thought*. Dorset House, New York, 1995.
- [25] Taligent Inc. Building object-oriented frameworks. a taligent white paper. <http://www.taligent.com/Technology/WhitePapers/BuildingFwks/BuildingFrameworks.html>, 1994.
- [26] Taligent Inc., editor. *The power of frameworks - For windows and OS/2 developers*. Addison-Wesley, 1995.
- [27] ISO/IEC. Information technology - process assesment. parts 1-5. IS 15504, 2003-2006.
- [28] H. Jaakkola and B. Thalheim. A framework for high quality software design and development: A systematic approach. *IET Software*, pages 105–118, April 2010.
- [29] R. Johnson and B. Foote. Designing reusable classes. *Journal of Object-Oriented Programming, SIGS*, 1(5):22–35, Jun/Jul 1988.
- [30] M. Klettke. *Modellierung, Bewertung und Evolution von XML-Dokumentkolektionen*. Advanced PhD (Habilitation Thesis), Rostock University, Faculty for Computer Science and Electronics, 2007.
- [31] T. Lewis, editor. *Object-oriented application frameworks*. Manning Publications Co., 1995.
- [32] Hui Ma, R. Noack, K.-D. Schewe, and B. Thalheim. Using meta-structures in database design. *Informatika*, 34:387–403, 2010.
- [33] Hui Ma, K.-D. Schewe, and B. Thalheim. Modelling and maintenance of very large database schemata using meta-structures. In *UNISCON*, volume 20 of *Lecture Notes in Business Information Processing*, pages 17–28. Springer, 2009.
- [34] D. Marco and M. Jennings. *Universal meta data models*. Wiley Publ. Inc., 2004.
- [35] M. Mattsson and J. Bosch. Framework composition: Problems, causes and solutions. <http://www.ipd.hk-r.se/michaelm/papers/frwkcomp.ps>, March 1997.
- [36] G. L. Murphy. *The big book of concepts*. MIT Press, 2001.
- [37] R. Noack. *Pattern for Screen and Interaction Design*. PhD thesis, Christian-Albrechts University Kiel, 2013.
- [38] S. Puroo and V. C. Storey. Intelligent support for retrieval and synthesis of patterns for object-oriented

- databases. LNCS 1331, pages 30–42, Los Angeles, USA, Nov. 3 - 5, 1997, 1997. Springer, Berlin.
- [39] K. Quibeldey-Cirkel. *Design patterns*. Springer, Berlin, 1999.
  - [40] G. Schellhorn. ASM refinement and generalizations of forward simulation in data refinement: A comparison. *Theor. Comput. Sci.*, 336(2-3):403–435, 2005.
  - [41] D. C. Schmidt. Applying design pattern and frameworks to develop object-oriented communication software. In P. Salus, editor, *Handbook of Programming Languages, Volume 1*. MacMillan Computer Publishing, 1998.
  - [42] D. C. Schmidt, R. E. Johnson, and M. Fayad. Special issue on patterns and pattern languages. *CACM*, 39(10), Oct. 1996.
  - [43] L. Silverston. *The data model resource book. Revised edition*, volume 2. Wiley, 2001.
  - [44] L. Silverston, W. H. Inmon, and K. Graziano. *The data model resource book*. John Wiley & Sons, New York, 1997.
  - [45] G. Simsion and G.C. Witt. *Data modeling essentials*. Morgan Kaufmann, San Francisco, 2005.
  - [46] E. Stabler. Derivational minimalism. In C. Retore, editor, *Logical aspects of computational linguistics*, volume LNCS 1328, pages 68–95. Springer, 1998.
  - [47] T. Stahl and M. Völter. *Model-driven software architectures*. dPunkt, Heidelberg, 2005. (in German).
  - [48] B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.
  - [49] B. Thalheim. The person, organization, product, production, ordering, delivery, invoice, accounting, budgeting and human resources pattern in database design. Technical Report Preprint I-07-2000, Brandenburg University of Technology at Cottbus, Institute of Computer Science, 2000.
  - [50] B. Thalheim. Towards a theory of conceptual modelling. *Journal of Universal Computer Science*, 16(20):3102–3137, 2010. [http://www.jucs.org/jucs\\_16\\_20/towards\\_a\\_theory\\_of](http://www.jucs.org/jucs_16_20/towards_a_theory_of).
  - [51] B. Thalheim. The theory of conceptual models, the theory of conceptual modelling and foundations of conceptual modelling. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 12, pages 543–578. Springer, Berlin, 2011.
  - [52] B. Thalheim. The science and art of conceptual modelling. In A. Hameurlain et al., editor, *TLDKS VI*, number 7600 in LNCS, pages 76–105. Springer, Heidelberg, 2012.
  - [53] B. Thalheim. Syntax, semantics and pragmatics of conceptual modelling. In *NLDB*, volume 7337 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2012.
  - [54] B. Thalheim. The definition of the (conceptual) model. In *Proc. EJC 2013*, pages 256–269, Nara, Japan, 2013.
  - [55] B. Thalheim and Q. Wang. Towards a theory of refinement for data migration. In *ER*, volume 6998 of *Lecture Notes in Computer Science*, pages 318–331. Springer, 2011.
  - [56] P. Wisse. *Metapattern - Context and time in information models*. Addison-Wesley, Boston, 2001.
  - [57] J. A. Zachman. A framework for information systems architecture. *IBM Systems Journal*, 38(2/3):454–470, 1999.

# GIS-Cloud Requirements Framework for E-Government Services

Fahdah F. AlOthman, Weaam F. AlMazyad, Ajantha Dahanayake<sup>1</sup>

*<sup>a</sup>Prince Sultan University – College for Women, King Abdullah Road, Riyadh 11586 Saudi Arabia*

**Abstract.** Various E-government services utilize GIS as solutions across disciplines, in order to tackle the complex problems. The E-government services have an initiatives to use the architectures of cloud computing, applications and platforms in order to deliver services and to enable them to meet the requirements and demands of their citizens. Cloud computing technology plays a vital role in addressing the challenges and issues which may occur in the applications of GIS. This research will discuss the use of GIS-Cloud in Saudi government, by examining and comparing the existing GIS application service in Qatar. By exploring the level of understanding of the GIS services, risk of data exposure when implementing these GIS-Cloud services, a requirements framework for GIS-Cloud in Saudi Arabia for E-government services is introduced.

**Keywords:** E-Government, Cloud computing, GIS, Framework, Requirement, Services, NIST.

## Introduction

E-government refers to the use by government agencies of information technologies to transform relations with citizens, businesses, and other arms of the government [1]. The government cloud computing initiative provides government agencies with services of high efficiency, reliability and security with respect to infrastructure, platform, and software [2]. The National Institute of Standards and Technology (NIST) provide an inclusive definition for the Cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interactions [3]. The Saudi E-government Second Action Plan includes establishment of a government cloud computing platform in order to deliver shared services to government sectors, as one of their future services is GIS.

GIS is computer software that links geographic information (where things are) with descriptive information (what things are). Unlike a flat map, where what you see is what you get, GIS can present many layers of different information [1]. GIS plays an essential role in wide range of areas and is extensively adopted nowadays. In the

---

<sup>1</sup> Corresponding author. Tel.: +966-11-494-8319.

*E-mail address:* ADahanayake@pscw.psu.edu.sa

simplest terms, GIS is the merging of cartography, statistical analysis, hardware, software and data. GIS is commonly used as a supporting system for making best possible decisions through spatial and non-spatial data relations, visualization and processing. GIS is beneficial and works well when made available to as many people as possible everywhere and anytime at the expense of very less resources in terms of technology and expenditure. Over a few decades efforts are being made to upgrade the conventional GIS applications in order to provide broad spectrum of services to the users across the globe.

Therefore, in this study we examine: What are the requirements for providing a new framework for E-government services using GIS coupled with Cloud computing?

## **1. E-Government, Cloud Computing and GIS**

### *E-Government*

E-government is the term used in the utilization of information technology by the agencies of government. E-government services may include mobile computing, internet technology, wide area networks etc. [4]. These information technologies are capable of enhancing and improving the relations of the government authorities with the government agencies, businesses, and citizens enabling the government to deliver improved and enhanced services to the citizens. The resulting advantages of E-government services may include cost reductions, revenue growth, greater convenience, increased transparency, and less corruption [4]. The processes associated with governmental services to citizens are considered as most time consuming process, but can easily be avoided by utilizing communication and information technologies. These technologies can serve a variety of different ends: better delivery of government services to citizens, improved interactions with business and industry, citizen empowerment through access to information, or more efficient government management. The resulting benefits can be less corruption, increased transparency, greater convenience, revenue growth, and/or cost reductions.

Traditionally, the interaction between a citizen or business and a government agency took place in a government office. Analogous to e-commerce, which allows businesses to transact with each other more efficiently (B2B) and brings customers closer to businesses (B2C), E-government aims to make the interaction between government and citizens (G2C), government and business enterprises (G2B), and inter-agency relationships (G2G) more friendly, convenient, transparent, and inexpensive.

For example, SADAD Payment System was established by the Saudi Arabian Monetary Agency (SAMA) to be the national Electronic Bill Presentment and Payment (EBPP) service provider for the Kingdom of Saudi Arabia (KSA). The core mandate for SADAD is to facilitate and streamline bill payment transactions of end consumers through all channels of the Kingdom's Banks [2].

### *GIS*

A geographic information system (GIS) can be defined as an organized collection of software and geographic data that allow efficient storage, analysis, and presentation of spatially explicit and geographically referenced information. A GIS provides a powerful analytical tool that can be used to create and link spatial and descriptive data for problem solving, spatial modeling, and presentation of results in tables or maps.

GIS data generally consist of two components: (1) Graphical data about geographic features (e.g. rivers, land use, political boundaries) (2) Tabular data about features in

the geography (e.g., population, elevation, modeled ambient concentrations of air toxics). GIS combines these different types of data using a “layering” technique that references each type of data to a uniform geographic coordinate system. Layered data can then be analyzed using special software to create new layers of data [5]. GIS is a software system that connects geographic information along with spatial information. GIS offers the integration of various layers of different kinds of information.

**Benefits of GIS for the government:** GIS can provide a variety of benefits for both the government and the citizens if it was used properly. GIS may enable the government of Saudi Arabia to increase efficiency of their operations while reducing the overall cost of those operations. These systems may also enable them to improve and enhance the coordination and the process of decision making. When the government of Saudi Arabia adapts new technologies to improve its E-government services, it can get a higher ranking in The "United Nations E-government Survey" which shows the ranking of countries that have put in place E-government initiatives and information and communication technologies applications for the people to further enhance public sector efficiencies and streamline governance systems to support sustainable development [6].

**GIS benefits for the society:** The most obvious benefit of GIS to the public is the ability to find all the needed and latest spatial and non-spatial information and analyze to meet their needs and these information can be trusted since its coming from the government. As well as delivering of services to citizens is achieved regardless of the boundaries between local government areas.

#### *|Cloud Computing:*

Cloud computing provides collaboration and greater efficiencies to the enterprise all across the world. Cloud Computing is often equated with the concept of a utility, in which an organization can “plug-in” to this virtual computing environment and use the computing resources available on an as-required basis [7]. Applications running on such a platform can be accessed via web clients, while the application software and data are kept at the (virtual) server side. A scenario is that components of an application are dynamically selected from a pool of services, and their coordination and computation are carried out at the client side, in the cloud, or both. Consistency in using various intellectual property (IP) rights, private data, ownerships of data of different clients and components intermix with the “distributed” program executions, which may be deeply embedded all over the cloud [8]. Conceptually Cloud Computing can be perceived as having five key characteristics (on-demand self-service; ubiquitous network access; location-independent resource pooling; rapid elasticity; and pay-per-use), three delivery models (SaaS – software as a service, PaaS – platform as a service, IaaS – infrastructure as a service), and four deployment models (private, community, public, hybrid) [3].



**Public Cloud:** Is available to the general public they can store their applications or data on shared servers are owned by an organization that selling cloud services. Using a public cloud model can give flexibility and cost saving, many feel can't trust on security of personal data stored on servers shared with other [9].

**Private Cloud:** Is used when organizations deploy cloud infrastructure inside them and managed by themselves. Costly when deploy internal cloud, the organization is in charge of monitoring and maintaining the data, ensure high levels of security and performance [9]. A private cloud shares many of the characteristics of public cloud computing including resource pooling, self-service, and elasticity and pay-by-use delivered in a standardized manner with the additional control and customization available from dedicated resources [10].

In order to collect data for this research project, two interviews were conducted. The first interview was conducted with the YESSER E-government program. This program handles the implementation of E-government services in Saudi Arabia and is controlled by a higher supervisory committee composed of minister of finance, ministry of communications and information technology (MCIT), and the governor of communication and information technology commission (CITC) [2]. The second interview was conducted with the ministry of municipality & urban planning- state of Qatar. These interviews played a vital role for obtaining appropriate, effective and complete information of the exiting GIS applications.

## 2. The Status of E-Government Service in Saudi Arabia

YESSER E-government program was a great source for getting the information. According to one of the YESSER's project managers who handled IT projects in different fields, Saudi E-government services have been improved tremendously in the few past years. The Saudi National portal includes 1554 e-services for more than 155 governmental agencies. Many government agencies provided their services through multiple channels such as mobiles and/or tablets [11][12].

The Saudi E-government Program (YESSER), the official representative of the E-government has been conducting e-transformation surveys for government agencies yearly. The total number of e-services identified is 4900 e-services (unrevised). There are many rules and policies regarding this aspect where general rules for data exposure or in certain levels for specific products like GSB (Government Services Bus). Ministry of Culture and Information have set rules for data exposure as they are responsible for data and information exposure in general. YESSER as the E-government program has adopted some regulations and decrees that contain rules and policies for Data Exposure. Government authorities shall take all necessary actions to guarantee this right for the service user/beneficiary [2].

GIS is used by different E-government services and the information required and processed is categorized as data using the following three categories: Human, Entity and Place. For example birth certificate: Human (Name, Certificate number,..), Entity (Hospital), and Place (The geo location "city").

Therefore, GIS is used whenever a service or an e-service requires an identification of a place. YESSER have also started an initiative to build a government cloud to serve government agencies and national applications. The private government cloud controlled by the environment will be implemented with consideration of the multiple usage, critical security, policies, and regulations for data exposure.

### 3. GIS Qatar

Qatar is the first country pioneered 20 years ago, and implemented a comprehensive nationwide GIS and is internationally recognized as having one of the finest GIS implementations in the world. Qatar established a National GIS Steering Committee and The Centre for GIS. The role of The Centre is to implement GIS in Qatar in an organized and systematic fashion and impartially serve the GIS requirements of all government agencies [13][14][15]. GIS was introduced to E-government about 5 years ago when CGIS (Center for GIS) assisted ICT in setting connecting to Qatar's GISnet and provided data for interactive mapping application on Qatar E-government website.

Qatar authorities set the objectives for a nationwide GIS and some of their approaches include [14]: (A) Eliminate duplication of efforts causing wastage of resources by avoiding data redundancy and by enhancing inter-agency co-ordination (B) Make right information available at the right time to the decision makers for efficient planning and management. (C) Foster teamwork among government agencies, especially those involved in physical and infrastructure planning, environment protection and local government administration so that they all work towards common goals (D) Efficiently manage government expansion for future development requirements (E) Achieve consistency and uniformity in policies, standards and regulations for whole of Qatar (F) Enable preparation of physical plans that are dynamic, flexible, easy to update, monitor and implement.

The implementation of GIS in Qatar was done using the following [14]: (A) Gaining support and commitment from the highest levels of government (B) Make concerned agencies and officials aware of the potential and power of GIS (C) Encourage government departments for co-ordination and data sharing (D) Involve every government department in design and implementation (E) Establish education and training programs and make GIS tools available to everybody.

The most important task of the GIS center was the development of digital mapping specifications and standards for the production of Qatar's Digital Topographic Database [14]. Furthermore, a computerized monitoring system of digital map was applied in order to enhance and improve the functionality and accuracy of the digital topographic database.

CGIS has also worked with all agencies that have applied GIS, in order to provide appropriate and proper guidance. It has also helped in developing and establishing the data dictionaries and specifications which are appropriate for their particular disciplines by enabling them to recognize the compatible and highly integrated standards and specifications of other agencies. This practice has resulted in highly developed and advanced data dictionaries and database specifications of national GIS. The main objective of such data dictionaries is to evidently demonstrate the type of data that is incorporated in its database. These data dictionaries also help in understanding the entire structure as well as the assembling of information. The method has enabled the government to ensure that each and every information or data, utilized by each agency is authorized. Moreover, this system has also allowed the government to assure that only a single agency is responsible for sharing, maintaining and collecting particular items of data. This system enables the government and organizations to reduce redundancy while improving programming, inventory management, field services, communication and quality of the services and products, provided to the citizens or users [16].

**Geodetic network:** The Centre for GIS is the custodian of Qatar's geodetic network that consists of approximately 6,000 horizontal control survey monuments and 4,500 vertical control survey stations distributed throughout the country.

**QCORS network:** is a Coordinated Network of nine Reference Stations distributed throughout the country that can provide Global Navigation Satellite System (GNSS) carrier phase and code range measurements in support of 3-dimensional positioning activities throughout the country which are an improvement of classical GNSS positioning done with a pair of GNSS receivers which commonly referred to as GPS.

**Qatar's Topographic Database:** the Digital Topographic Database is comprised of four separate components, which have been designed and developed to meet all known GIS application needs of government. These components are: (A) the up-to-date Digital Topographic Base Maps. (B) High precision, Digital Elevation Model (DEM). (C) Orthoimagery. (D) High resolution satellite imagery and four directional oblique aerial imagery.

CGIS also maintains and updates Qatar's geographic names and landmarks database.

**Regional Training:** As a result of the world prominent position held by Qatar's GIS, the Centre for GIS reached an agreement with Environmental Systems Research Institute (ESRI) to establish an authorized regional training program [17].

**GIS Applications:** GIS services are provided to any government agency upon request. CGIS provides training, technical support and GIS software licenses to the respective agency free of charge. As for private agencies, certain data sets are sold on an individual bases and recently they have started providing web based GIS services to some government and private agencies which are conducting projects to the government.

Some of the applications of GIS in an agency are described below:

**Web Services:** provide better and enhanced opportunities to the users to access and apply GIS, with in their working environment. The web portal serves as a comprehensive reference for a wide range of electronic mapping services [17].

**Mobile Apps & Services:** Al Murshid is considered as one of the famous and leading mobile applications in Qatar. Al Murshid provides several useful functions through a simple interface in both Arabic and English language. Their functions are: (A) Locating any of the 6000+ landmarks collected and maintained by CGIS. (B) Users can use the "nearby" function to perform spatial (C) Routing functions between two or more points. (D) Qatar Area Referencing System

The project has created a great deal of enthusiasm and potential for a wide range of practical applications in Qatar. The applications include municipalities, water, electricity, drainage, routing, telecommunications, emergency, statistics, education, postal and delivery services [17].

#### 4. GIS Cloud Requirement Framework

According to the Esri President Jack Dangermond, Esri deals with the variations which are expected in the GIS [18]. GIS is swiftly changing its functions and operations, with the passage of time. Furthermore, he has also claimed that, it is expected that GIS will be able to be used everywhere, regardless of the location and time. In addition to this, the collaboration, neogeography and crowd sourcing are the most prominent variations in the technology of GIS. This study has utilized an approach of multi-tiered architecture, which divides diverse logical features of cloud systems of GIS, in order to develop and enhance the abilities of every single component. In addition, this system may offer elastic platforms, extensive business intelligent systems, personalized and

secured environments, scalable (vertical and horizontal) infrastructures, heterogeneous platforms and flexible solutions to the users of GIS [19]. GIS has proved to be a flexible, adaptive technology, evolving as the ecosystem around it. At each step in this evolution, GIS has not just adapted to these changes but embraced them, becoming more powerful and more valuable.

The requirements framework is derived by integrating the findings from Esri and the interviews conducted with the Yesser E-government program and GIS Qatar. The requirements framework for GIS-Cloud E-government services in Saudi Arabia is made up two categories, technical and non-technical requirements:

Technical	Strategy, Planning, Policy, and regulations
<ul style="list-style-type: none"> <li>• Creating and updating Saudi topographic database.</li> <li>• Maintaining Saudi’s geodetic network and all related services. These services and networks may include geodetic network, global navigation satellite system (GNSS) and GNSS receivers (commonly referred to as GPS).</li> <li>• Establishing and maintaining Saudi’s high speed GIS data sharing network (GISnet), linking it with agencies GIS databases and securing its data.</li> <li>• Providing the connection to Saudi’s topographic database and maintaining it.</li> <li>• Creating, developing and monitoring national GIS standards and specifications.</li> <li>• Providing technical assistance.</li> <li>• Implementing GIS for all ministries and government agencies.</li> <li>• Analyzing the requirements of GIS software licenses.</li> <li>• Providing GIS e-services for Saudi.</li> <li>• Managing and administering Saudi’s GIS infrastructure.</li> <li>• Architecture of GIS Cloud may include [20]:               <ul style="list-style-type: none"> <li>• GIS cloud configuration layer.</li> <li>• GIS Cloud Logic Layer.</li> <li>• GIS Cloud Utilities Layer.</li> <li>• GIS Cloud Repository Layer.</li> <li>• GIS Cloud Communication Layer.</li> <li>• GIS Cloud Web-Interface.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Cooperating with Arab regional and international bodies associated with geographic information systems.</li> <li>• Designing and implementing plans for maintaining safety and security.</li> <li>• Training and education programs like seminars, workshops etc.</li> <li>• The entities of government will recognize their data and information in accordance to their combined guidelines of levels and specifications, set by programs of E-government. .</li> <li>• All the entities of the government will make plans to reduce repetition and duplication in their databases; ensure the integrity of information and data.</li> <li>• Each and every entity of the government entity will monitor the performance of their own databases and ensure the availability of their shared data electronically.</li> <li>• Assist planning and implementing electronic terms and conditions of the government services.</li> <li>• For the purpose of accuracy, the entities of the government will allow forth and pursue clear and specific mechanisms.</li> <li>• No data or information will be requested from the data applicant.</li> <li>• Only the authorized persons shall review information and data relevant to the government services and applications.</li> <li>• Appropriate and effective plans to reduce redundancy in the databases.</li> </ul>

**Table 1.** Requirements Framework of GIS Cloud for E-government Services in Saudi Arabia

## 5. Conclusion

Although there is much remains to be done, this study generated important findings on the requirements framework for E-government services for Saudi Arabia using a GIS-Cloud. However, we do confirm that there are some limitations to our study. As the fields of GIS, Cloud computing, and E-government are huge and requires more time to

research. The access of data is another limitation. Most of the information needed was not available to the public, especially information about e-government services in Saudi Arabia and Qatar. Interviews were conducted with the concerned people to get the information need for this research.

GIS is rapidly moving towards the vision of ubiquity through cloud computing. There are limitations to cloud computing due to the strict rules and regulations on the extent of data exposure. Security issues needs to be resolved in order to make the cloud computing efficient and protected. Therefore using both private and government cloud for the critical, sensitive data with GIS, and the public cloud for the insensitive, public data will be the best solution for the E-government services in Saudi Arabia.

## References

- [1] Esri, What is GIS, July 2012.
- [2] YASSER, <http://www.yesser.gov.sa>, retrieved on October 2013.
- [3] National Institute of Standards and Technology, Information Technology Laboratory, retrieved on October 2013.
- [4] A. AlAjeeli, Y.A. Al-Bastaki, Handbook of research on e-services in the public sector : E-government strategies and advancements, Hershey, PA, 2011.
- [5] M. Peggion, A. Bernardini, M. Masera, GEOGRAPHIC INFORMATION SYSTEMS AND RISK ASSESSMENT, European Communities, 2008
- [6] UN E-government Surveys, <http://www.unpan.org>, retrieved on November 2013.
- [7] L.M. Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner, A break in the clouds: towards a cloud definition, SIGCOMM Comput. Commun, 2009.
- [8] W. Vogels, A Head in the Clouds – The Power of Infrastructure as a Service, In First workshop on Cloud Computing and in Applications, 2008.
- [9] A. AlSaikhan, H. AlMajhad, L. AlHoraibi, THE ADOPTION OF CLOUD COMPUTING IN SAUDI'S ORGANIZATIONS, Al Imam Muhammad Ibn Saud Islamic University (IMSIU), 2013.
- [10] M. Amini, N.S. Safavi, S.M. Khavidaki, A. Abdollahzadegan, Type Of Cloud Computing (Public And Private) That Transform The Organization More Effectively, International Journal of Engineering Research & Technology (IJERT), 2013, 2278-0181.
- [11] B.A. Alsheha, The e-government program of Saudi Arabia Advantages and challenges. King Fahd University of Petroleum and minerals, Finance and Economics Department, 2007.
- [12] Saudi, [www.saudi.gov.sa](http://www.saudi.gov.sa), retrieved on October 2013.
- [13] QUIP, <http://quip.qatar.vcu.edu/ica-atom2/index.php/>, retrieved on November 2013.
- [14] Q.M. Al Ghanem, Qatar's GIS - A Unique Model for Next Millennium GIS, Esri, retrieved on November 2013.
- [15] S.W. Rouag, Information Technology in Qatar, IT Geographics, retrieved on November 2013.
- [16] X. Zheng, J. Zhu, Q. Yan, Monthly air temperatures over Northern China estimated by integrating MODIS data with GIS techniques. Journal of Applied Meteorology and Climatology, 2013.
- [17] CGIS, <http://www.gisqatar.org.qa/>, retrieved on November 2013.
- [18] J. Dangermond, GIS in a Changing World, Esri, 2010.
- [19] L. Dongrong, Using GIS and Remote Sensing Techniques for Solar Panel Installation Site Selection, 2013.
- [20] M.A. Bhat,R.M. Shah,,B. Ahmad, Cloud Computing: A solution toGeographical Information Systems (GIS), International Journal on Computer Science and Engineering (IJCSSE), 2011, 0975-3397.
- [21] iTunes, <http://www.apple.com/>, retrieved on December 2013.
- [22] Supreme Education Council, <http://www.sec.gov.qa>, retrieved on December 2013.

## Trust levels of Mobile Banking Apps

Ahlam Alshareed<sup>1</sup>, Hadeel Alsagyyer<sup>1</sup>, Hanieah Alenizi<sup>1</sup>, Ajantha Dahanayake<sup>1</sup>,

<sup>1</sup> Prince Sultan University, Dept. Of Computer Information Systems, Riyadh, KSA  
[adahanayake@pscw.psu.edu.sa](mailto:adahanayake@pscw.psu.edu.sa)

**Abstract:** Trust and security affect the usage of mobile applications. The aim of this research is to use trust as the primary component in mobile applications in banking sector. Accordingly, the level of trust in m-banking is researched from technical and usage perspectives. The data for this study has been collected using a survey of 168 samples of mobile applications users and interviews with banking application developers. Finally, the results of the surveys were integrated into a framework for identifying trust levels in m-banking in order to provide insights into the issues of trust and its relation to m-banking application development.

**Keywords:** M-banking, Mobile applications, Saudi Arabia, Security, Trust

### Introduction

Mobiles have begun to play an integral role in bank transactions. Online transactions are often accompanied by feelings of fear and anxiety among customers. The faceless and intangible nature of these transactions can affect the customers' willingness to engage in such banking activities. Central to this research is the concept of trust as users' acceptance of bank's mobile applications for transactions. The concept of trust is explored in order to gain insight into its relation to m-banking. Trust is explored through the definitions and concepts introduced in the literature which relate to customer satisfaction and cultural dimensions [20, 10]. The customers are the fundamental valuable sources for bank's revenue. Trust has been selected as it is the main factor affecting the acceptance of m-banking transactions that determines the quality of customer relationships [3].

The lack of trust poses a significant problem to a bank's financial success. During every transaction, the parties involved should feel the trust. It must be established and managed continuously in money transaction activities [9]. To ensure trust, some security services are offered to protect from security threats; identification, authentication, confidentiality, integrity, access control, and non-reputation are some of these examples [4]. Today, banking applications are doing more than ever to increase efficiency and improve relationships with customers. In relation to trust and internet technologies, customers tend to have two main concerns – privacy and security. This study focuses on “the trust as an imperative factor for using m-banking apps in Saudi Arabia and developing a framework to integrate those trust issues into the development of m-banking apps”.

### 1. Related Works

As mobile apps are a new technology, it may feel as if it is an upgraded version of e-services. In [5] investigate the relationship between perceived ease of use, perceived usefulness, perceived risk, social influence, and customer intention to adopt m-

banking within the context of the low-income population of Pakistan. The study looked into the familiarity of technology use and its effect on trusting mobile apps during banking transactions.

Another survey was conducted in 2013 by the Consumer Research Section of the Federal Reserve Board's Division of Consumer and Community Affairs (DCCA) in United States Washington DC. The study examined the consumers' use of mobile financial services including m-banking. The adoption of m-banking had increased substantially over the past year. Nearly 28% of mobile phone users surveyed used m-banking in the past 12 months, a significant increase from 2011. The two factors limiting consumer adoption of m-banking and payments are; the security concerns of the technology and a sense of not offering any real benefits over the existing methods for banking or making payments.

Today, the consumers are bold enough to report that they simply do not know how safe it is to use m-banking [6]. This finding suggests that consumers need to be provided with reliable and accurate information on security associated with the various means of accessing m-banking. In terms of value proposition to consumers, the significant number of mobile users who reported interest in using their phones to receive discounts, coupons, and promotions or to track rewards and loyalty points suggest that tying these services for mobile payment services would increase the attractiveness of mobile phones as a means of payment [6].

Most users who do not trust m-banking think that security is the key factor. F-Secure Company presented a Mobile Threat Report Q4 2012 [13] that clearly identified that Android malware has been strengthening its position in the mobile threat scene. Every quarter, malware authors bring forth new threat families and variants to lure more victims and update existing ones [7]. Unfortunately, literature does not pay much attention to trust and trust levels associated with m-banking apps.

## **2. The Framework**

The framework presented in this research seeks to define those factors that identify the trust issues and then to understand the meaning of trust from the customer's perspective. It relates trust to the technology and regulations that help to increase users' trust in m-banking applications. The figure 1 represents the framework for assessing factors that influence trust in m-banking, and this framework further explained in this section.

### **2.1. Why Trust Matters?**

Trust is normally specified in terms of a relationship between a trustor, i.e., the subject that trusts a target entity, and the trustee, i.e., the entity that is trusted. Each trust relationship must be defined with respect to a particular scenario or context, can be viewed as a mathematically defined binary relation, must have an associated trust level, can be stated as adhering to some property, and is influenced by auxiliary factors [8].

According to our view trust is "the quantified belief of a trustor with respect to the competence, honesty, security and dependability of a trustee within a specified context". 'Truthfulness' refers to the state wherein one consistently utters what one believes to be true. A secure entity ensures the confidentiality of its valuable assets and prevents unauthorized access to them. Dependability measures the extent to which reliance can justifiably be placed on the service delivered by a system [8].

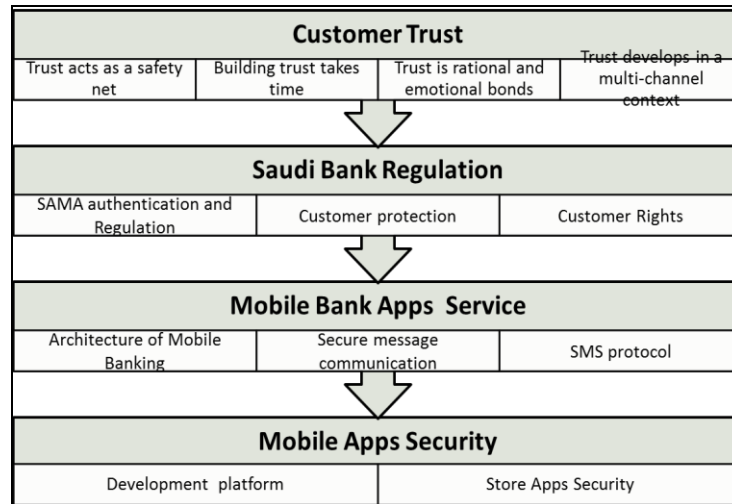


Figure 1: Framework for accessing the factors that influence trust levels in m-banking

### M-banking applications

M-banking can be subdivided into three key areas: Informational; Transactional; and Service, Marketing & Acquisition. Within the informational area, there are functions such as balance and transaction history, loan, mortgage, and credit information, ATM and branch locators, as well as personal financial management (PFM) functions such as peer spending comparisons and budget tools. Transactional services included account transfers, bill pay, person-to-person payments, and remote deposit capture. Service features include functions that enhance the customer's experience; including contact options, help information, and alerts. Additional service features include product renewal notifications, balance-triggered savings offers, balance-triggered credit offers, and location-triggered travel insurance options. Relative to marketing and acquisition, services are offered such as mobile coupons/incentives, barcodes, new product information, customer research, cross-selling, and acquisition. The aspects of m-banking that make it particularly appealing to marketing are the very personal nature of mobile devices and the "always on" aspect for customer use [11].

### How Does The Customer View Trust?

From the literature on customer's experience, relationships, the role of emotions and trust [12] the following points of general consensus are distilled:

**-Trust acts as a safety net.** In situations of perceived risk or vulnerability, trust plays the role of a safety net, helping the customer to make a clear decision by minimizing uncertainty and risk.

**-Building trust takes time.** Trust develops in stages on the basis of a gradual deepening of the relationship and mutual adaptation to the needs of the other party, so trust emerges from the accumulation of satisfactory previous experiences.

**-Trust is created through both rational and emotional bonds.** 'Rational trust' refers to the customer's willingness to rely on a service provider's competence and reliability. 'Emotional trust' is the confidence arising from the customer's feelings generated by the level of care and concern the other party demonstrates.

**-Trust develops in a multi-channel context.** Trust is most often associated with the overall organization as the main target of trust; however, in today's multi-channel service environment, emotional and rational bonds of trust are created with multiple "agents" or touch points,



including the front-line staff, the self-service technology (e.g., ATM, e-commerce, online account management) and an increasingly complex array of marketing communications.

Trust is based on evaluations of 3 complementary dimensions: competence, or credibility; integrity, or honesty; and empathy, or benevolence. The second and third of these can be interpreted as being more 'emotional trust', while the first is more 'rational trust' [12].

### **2.1 The Evolution of Self-service Banking Channels:**

While in-person transactions are declining, mobile and Web services are growing. Research shows that the phone channel (call center + IVR) has remained steady since 2007 with a slight upward trend forecasted to 2015. The phone channel is the stalwart among all banking channels; other channels rise and fall, but the phone channel remains steady. Even though it may be overshadowed by growth in internet and mobile, the phone channel is not going away any time soon [13]. By 2015, the phone channel will support more transactions than ATMs or branches [13].

**M-banking:** Although this channel is relatively new, it is already showing steady growth. Used in its early stages as a push/pull tool for information text messages, cell phone banking now supports personal account access and is forecasted to become the new mobile payment method or "digital wallet" of the future [13].

This year's distribution of primary channel use was as follows [14]: (a) Internet Banking (laptop or PC) – 39% (36% in 2010), (b) Branches – 18% (25% in 2010), (3) ATMs – 12% (15% in 2010), (4) Mail – 8% (8% in 2010), (5) Telephone – 9% (6% in 2010), (6) Mobile (e.g., cell phone, Blackberry, PDA, iPad) – 6% (3% in 2010).

According to [14] "These results show customers are embracing new technologies that make managing a bank account simpler, easier and more convenient, but that doesn't mean that the traditional bank branch is going anywhere soon. Branch design may evolve as a result of declining foot traffic. However, we know that nothing replaces human interaction and that's why branches will never disappear."

Going forward, the key will be to encourage adoption of the mobile channel by more than just the early adopters for more robust use beyond simple balance checking. According to the white paper [14]: To surpass this 'tipping point', [14] recommends five factors that will help move m-banking into the mainstream: (1) Establishing m-banking as useful, (2) Providing access to m-banking through all devices, (3) Helping consumers overcome security concerns, (4) Fostering familiarity to facilitate a natural transition across channels, (5) Making m-banking easy to use.

### **Saudi Bank Governance**

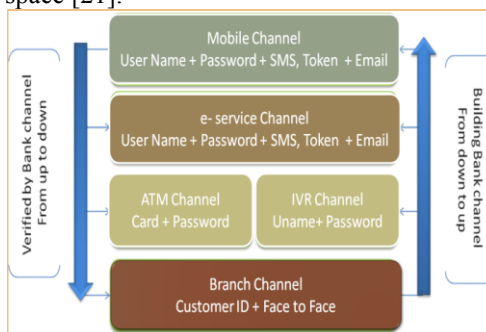
The Saudi Arabian Monetary Agency (SAMA) [15], which is the central bank was established in 1372H (1952). It has been entrusted with performing many functions pursuant to several laws and regulations.

There are two important regulations that cover customer rights - E-Banking rules and Consumer Protection Act. They explain banks' responsibility of customer protection from different perspectives – security, confidentiality, and awareness.

SAMA published the "Legal and Reputational Risk Management in E-banking Rules" under Principles 11-14. The M-banking service available for customers in Saudi Arabia are a limited subsection of banking services: Account (View last 5 transaction and Full statement), Credit cards (Card details, Unbilled transaction, Card statement, Card payment), Transfer under 40,000 SR (Between your bank accounts, To other bank accounts, To local banks, To international banks, To international banks), Utility Bills (pay bills, Add/remove bills).

### M-banking development apps vs. Trust

A key area of concern for consumers and financial service providers is the security of m-banking and payments. In addition to the newness of the technologies and entrants, there is a complex supply chain that increases the chance of security risks. As of yet, there is no real standard for technology that has captured the market, and regulations regarding some of the new entrants are non-existent. Customers have increased control of their device in terms of application downloads, OS updates and personalization of their devices. This will lead to new challenges regarding privacy, and it will take some time before the younger generation realizes the implications of privacy violations. Compounding the challenge is the fact that traditional security controls, such as firewalls, and encryption, have not reached the level of maturity needed in the mobile space [21].



**Figure 2:** The relation between building new channel and verify process in Saudi Arabia.

### Development platform

To build an application, a developer must be familiar with the device's programming language. The developer will also need access to the device software developer kit (SDK), which in turn gives the developer access to the device's application programming interface (API). The SDK includes several tools, including sample applications and a phone emulator. Emulators are programs that duplicate the features and functions of a specific system or device. When developers finish building their applications, they can test them out on the emulator to see how the app will perform on actual hardware. The Worldwide Smartphone Market Continued to Soar, in 2012. Windows Phone 8 was launched in Fall 2012 [22].

### Architecture of M-banking

The architecture of m-banking has layers that increase the security of mobile transactions. M-bank Management includes two parts – M-banking Gateway and M-banking Service – that connect with a Bank Datacenter within the domain of the bank organization. While the mobile application calls the m-banking service through 2 layers – Firewall, which is a physical layer, and encryption of messages [16].

### Securing Message Communication of Mobile Web Services

At the minimum, mobile Web service communication should possess the basic security requirements of proper authentication/authorization and confidentiality/integrity. Secure message transmission is achieved by ensuring message confidentiality and data integrity, while authentication and authorization ensure that the service is accessed only by the legitimate service requestors [16].

To achieve confidentiality, the Web service messages were ciphered with symmetric encryption algorithms and the generated symmetric keys were exchanged by means of

asymmetric encryption methods. The messages were tested against various symmetric encryption algorithms, along with the WS-Security mandatory algorithms. The public key infrastructure PKI algorithm used for key exchange was RSA-V1.5 with 1024 and 2048 bit keys. Upon successful deployment of confidentiality, data integrity is considered on top of confidentiality [16].

#### **Factor of Authentication Used in Mobile Phones**

**Authentication based on something the user knows:** Authentication is based on confidential information that only the user knows and is mostly used as a means of authentication for mobile services. The user is usually prompted just for username and password. The user is generally allowed to choose his/her passwords in order to greatly enhance the security of his/her private information. Also, most services offer users the option to save the password locally on the device and when accessing the most popular mobile services (i.e., Internet browsing, mail), that way influence the security or usability of the service according to his/her needs [17].

**Authentication based on something the user has:** For this kind of user authentication, the user is required to be in possession of a specific physical object – namely, an authentication token. There are different types of tokens, and they can vary from cards with printed passwords that require the user to retype password to specific devices that can be connected to the user's terminal. When tokens are used with a mobile device, one very good approach is to store private information on the mobile device and then use the mobile device at the same time as the token and terminal. This is an example of how the specific characteristics of a mobile device can be utilized to enhance the usability of an authentication method. Using this approach, there are three different ways in which specific information can be actually stored on the device: on hardware of the mobile device, in a specific file in the memory of the device, or on the operator's side. In Norway, a couple companies are providing solutions that utilize a token integrated with the mobile device. A very short description of what is describes in [17].

How the Secure SMS protocol conforms to the general security requirements.

**Confidentiality.** This is achieved by encrypting the message through a symmetric secret one-time password. This one-time password is only shared between the user and the bank server [5].

**Integrity.** The message digest is the hashed value of the message content calculated server application and the mobile phone application. If the content is altered during transmission, the hashing algorithm will generate a different digest value at the receiver side [5].

**Authentication.** For the receiver to authenticate the user, the user must provide his/her authentication detail(s) to the receiver. This authentication process is performed by validating the message PIN with the receiver-stored PIN. The PIN was previously selected by the user when the user registered for a m-banking account. The strength of the authentication depends on the password selection strategies used [5].

**Non-Repudiation.** Only the account holder and the bank server are supposed to have the one-time password. The bank server does not generate the same password more than once [5].

**Availability.** The availability of this protocol depends on the availability of the cellular network. The time it takes for a message to be delivered depends on the density of network operator base towers. The number of transactions that the server can handle at any one time depends on the hardware capability. If the server hardware can handle multiple incoming messages, then the server can perform multiprocessing to accommodate more requests. The protocol has no restrictions on the type of hardware needed. Therefore, it is up to the developers to decide the hardware specifications [5].

### **Security of Application Store\Market**

Apple conducts identity verification to authenticate individuals or company's identity and eligibility to enroll in an Apple Developer Program. Companies and educational institutions must provide a D-U-N-S (Data Universal Numbering System) Number registered to their legal entity as part of the enrollment process for Apple Developer Programs. An individual does need a D-U-N-S Number for enrollment [18]. In the android market there is no clear polices about the identity verification. They only mentioned on their side, "If you are an organization, consider registering a new Google account rather than using a personal account" [19].

### **3. Summary of Survey Data**

In reference to the survey, which covered 168 samples of Saudis, 98.1% have an account, and 98.4% use smartphones; 66.1% of the sample use m-apps while 33.9% do not. According to our findings, 74.5% are e-service users and 34.7% use m-banking and other channels. 95% know about m-banking service; this is due to SMS and email awareness. From a security point of view, 98.8% of prefer to receive SMS notification after each transaction; this feature helps to protect customers by keeping them up-to-date.

An interview held with three bank developers regarding the level of trust of a bank customer mentioned that no complaints registered from any customer about the security issue; also the development depends on securing the messages between bank servers, web service, and mobile apps. The development code scans the smart phone before opening the apps to see if there is any spy system. The limitation of services is also described clearly by them; they said that, according to SAMA regulation, adding a beneficiary is not allowed because the mobile is subject to theft. According to an information technology bank manager, their statistics show that ATM is the most widely used channel; after that e-service then m-banking and the most important factor of trust comes from the bank reputation.

### **4. Conclusions**

The research presents a framework that integrates the factors that influence trust in m-banking applications development. Trust is a relationship between trustor and trustees; for bank customers. This relationship is built slowly by the bank customers due to customer trust and its complementary dimensions: credibility, integrity, and empathy. The second and third of these can be interpreted as "emotional trust"; which is built by SAMA regulations and bank security rules. SAMA helps to define and clarify customer rights and protections, as well as an umbrella that makes the customer feel protected by the government and bank security rules, while the credibility is "rational trust" and is built by mobile security of the connectivity and mobile platform. From the customer's point of view, the balance between benefits of accessibility and risk of security depends on their own choice. For Saudi mobile users, according to the existing studies, the m-banking services will further evolve in the next three years by more than 30%. Therefore, the bankers should spend more efforts to increase the security of m-banking apps and increase the awareness of banking channels and its robustness.

Any new banking service channels needs to be built on the previous channels. The relationship between building new channels and verification process will help to inherit the levels of trust of previous channels and increase it in the new channels. Saudi culture is very different from western, Asian, or other moderate middle-eastern cultures [10, 20]. Therefore, trust is builds over a long time span [1]. The main shortcoming of this research is that we have not investigated how trust is built in different culture dimensions [10, 20]. One of the future research directions will be to study trust from different cultural perspectives and how trust influences the m-banking application along different cultural tapestries [1].

### References

1. R.D. Lewis. *When Cultures Collide. Managing Successfully Across Cultures*. Nicholas Brealey, London, 3rd edition, 2011.
2. Google Inc. *Our Mobile Planet*, [Online] available at: <http://www.google.com/think/research-studies/our-mobile-planet-saudi-arabia.html>, Sep 17, 2013.
3. Wang, Y.S., Wang, Y.M., Lin, H.H., & Tang, T.I., *Determinants of user acceptance of Internet banking: An empirical study*, (2003).
4. Prof. Dr. Eldon Y. Li. *International Journal of Electronic Business Management*, Vol. 7, No. 3, pp. 151-158 ,(2009).
5. Kelvin Chikomo, Ming Ki Chong, Alapan Arnab, Andrew Hutchison. *Security of M-banking*, University of Cape Town, November 11, 2006.
6. Kazi, Abdul Kabeer and Mannan, Muhammad Adeel. *Factors affecting adoption of m-banking in Pakistan: Empirical Evidence*. Published in: *International Journal of Research in Business and Social Science* , Vol. 2, No. 3 (15. July 2013): pp. 54-61.
7. Fedral. *Consumers and Mobile Financial Services*, March 2013.
8. F-Secure. *MOBILE THREAT REPORT Q4*, 2012.
9. Tyrone Grandison and Morris Sloman .*Trust Management Tools for Internet Applications*. iTrust'03 Proceedings of the 1<sup>st</sup> international conference on Trust Management, pp 91-107, Springer-Verlag Berlin, Heidelberg, 2003.
10. G. Hofstede, G.J. Hofstede, and M. Minkow. *Cultures and Organizations: Software of the Mind: Intercultural Cooperation and Its Importance for Survival*. McGraw-Hill, New York, 2010.
11. Vanessa Pegueros. *Security of Mobile Banking and Payments GIAC* , November 1, 2012.
12. Prof. Chris Halliburton and Adina Poenaru .*The Role of Trust in Consumer Relationships*, 2010.
13. Adi Kohali ,Adi Sheleg . *Alternative Banking Channels*, Dec 2010.
14. Michael Flores. *Retail Banking Success: The Important Role of Optimized Interactive Voice Response (IVR) Systems*, Mar 11, 2013.
15. Saudi Arabian Monetary Agency (SAMA). *SAMA's Functions*. [Online] available at: <http://www.sama.gov.sa/sites/samaen/AboutSAMA/Pages/SAMAFUNCTION.aspx>, Dec 17, 2013.
16. Satish Narayana Srirama and Anton Naumenko. *Secure Communication and Access Control for Mobile Web Service Provisioning*, Jul 21, 2010.
17. Aloul, F. ,Zahidi, S., El-Hajj, W. *Two Factor Authentication Using Mobile Phones*, May 2009.
18. Apple Inc, *Identity Verification*. [Online] available at: <https://developer.apple.com/support/ios/identity-verification.html>, December 17, 2013.
19. Google Inc. *Developer Distribution Agreement*. [Online] available at: <https://play.google.com/about/developer-distribution-agreement.html>, December 17, 2013.
20. G. Hofstede. *Cultural dimensions - WWW*. <http://www.geert-hofstede.com>
21. Vanessa Pegueros. *Security of Mobile Banking and Payments*, November 1, 2012.
22. IDC. *Android- and iOS-Powered Smartphones Expand Their Share of the Market in the First Quarter*, May 24, 2012.

# Conceptual Framework for Big Data Analytics Solutions

Mashail ALSWILMI<sup>a</sup>, Nouf ALNAJRAN<sup>a</sup>, and Ajantha DAHANAYAKE<sup>b,1</sup>.

<sup>a</sup>*Prince Sultan University – College for Women, King Abdullah Road, Riyadh 11586 Saudi Arabia*

<sup>b</sup>*Tel.: +966-11-494-8319, E-mail address: ADahanayake@pscw.psu.edu.sa*

**Abstract.** This Big Data is generated fast and needed to be processed fast accordingly. It requires a new ways to analyze data that goes beyond leveraging incumbent tools. This revolution is associated with a challenge that lies not only in the collection and storing of massive and diverse amounts of data, but also analyzing and extracting value from this data. Big Data represents a revolutionary step forward from traditional data analytics, characterized by its four main elements: Variety, Volume, Velocity, and Veracity. This research highlights the importance of Big Data, and classifies Big Data problems according to the format of the data that must be processed, and presents a conceptual framework for mapping Big Data types and the appropriate combinations of data processing components – the processing and analytic tools to generate useful patterns for understanding Big Data analytics process.

**Keywords:** Big Data, Hadoop, MapReduce, YARN, HDFS, Big Data business problems.

## Introduction

As the amount of data in our world has been exploding, companies capture trillions of bytes of information about their customers, suppliers, and operations. Millions of networked sensors are being embedded in the physical world in devices such as mobile phones and automobiles, for sensing, creating, and communicating data. Multimedia and individuals with smartphones on social network sites will continue to grow exponentially. The type of data that is called Big Data, are large pools of data that can be captured, communicated, aggregated, stored, and analyzed is now part of every sector and function of the global economy [1]. Therefore, it is a reality dealing with this enormous growth of data, specifically the semi-structured (e.g., logs) and unstructured data (e.g., audio, video, and e-mail) in the order of terabytes or even petabytes. Dealing effectively with Big Data requires performing analytics against the volume and variety of data while it is still in motion, not just after it is stored. Therefore, in order to analyze Big Data, different analyzing approaches are needed for extracting value and opening up new opportunities that goes beyond the leverage of available incumbent tools which are costly and unable to meet the demands of the new “Big DataAnalytics” landscape.

In this context, Hadoop [2] as a Big Data processing open source framework has rapidly become the de facto standard in both industry and academia. The main reasons of such popularity are the ease-of-use, scalability, and failover properties [3]. Almost

---

<sup>1</sup> Corresponding Author.

all big software vendors such as IBM, HP, Oracle, SAP, or even Microsoft use this Hadoop framework.

The Apache Hadoop software library is a framework that allows the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures [4].

IBM grossed over \$ 1.3 billion from Big Data product stream covering server and storage hardware, analytics applications and database software. IBM's Big Data solutions have Hadoop based analytics, stream computing, data warehousing and information integration at the core, operating on a platform of visualization and discovery, application development, accelerators and systems management [5]. IBM is followed by HP with revenues of \$ 664 million, with similar mix of hardware, software and services packages for Big Data [5]. HP's Vertica analytics, Enterprise Edition Software which manages massive amounts of data quickly and reliably, gives real-time business intelligence for advanced, Big Data analytics [6], has given the company a boost in Big Data. HP offers end to end solutions based on open source, covering infrastructure to capture, store and scale Big Data through HP BladeSystems [5].

The main idea about this evolving IT trend in Big Data is to draw insights for situations that produce business value, such as customer behavior. The customer behavior is the most valuable information for almost every company. The customer satisfaction is a key factor that contributes to the company's return of investment (ROI). Such important information is produced through the analytics of these Big Data collections particularly collected according to the needs of specific domains. One of the main problems of Big Data analytics is the availability of vast amounts of analytic tools and the difficulty of finding or matching the right BD analytic tool to specific analytic need or situation. Secondly, most of the BD analytic tools are de facto data mining tools that belong to the generation of problem solving of structured data analytics. Therefore, in this research we concentrate on defining a requirements model for Big Data analytics tools through questioning: "What are the requirements of an analytical tool to conduct successful analysis of generated Big Data?"

This research introduces a conceptual model for mapping different types of analytical tools required to store, analyze, and process these huge data according to particular business problems. Thereby, bringing a better understanding into Big Data analytics tools by revealing which tools are equipped to handle data that traditional relational databases, data warehouses, and other analytics platforms have been unable to effectively manage.

The remaining of the paper is structured as follows: the next section will provide a glimpse on literature review of related works. The section two provides definitions of frequently used terms and concepts in this research. The section three is the proposed framework for identifying the analytical tools requirements for different classifications of Big Data business problems, with the solution along with the encountered limitations. The section four presents conclusions at the end of this research paper.

## **1. Literature Review**

Integrating analytical tools to solve Big Data problems is challenging. In [7], the authors discussed Big Data technology along with its importance in the modern world

and existing projects in order to efficiently and effectively extract value and transform to better domains. They have also discussed the Big Data analytics framework (Hadoop) in detail, along with the problems Hadoop is facing, and the Good Big Data practices to be followed. While, there are a lot of researches based on benchmarking, this particular research [8] uses benchmarking to study different analytical tools. The authors have produced a comparative study between two different types of Big Data analytical tools, the revolutionary enterprise analytical tools and the open source tools for the same process. They have covered many different enterprise and open source platforms for Big Data analytics and compared them based on computing environment, amount of data that can be processed, decision making capabilities, ease of use, energy, and time consumed, and the pricing.

Azemovic and Music in [9] emphasized on mechanisms of storing unstructured data. They have adopted an experiment strategy and comparative analysis to presented statistical results on using different methods for storing these data within database and classical file systems. They have discussed their advantages and disadvantages and came up with a model for testing and benchmarking system for storing unstructured data. In [10] the authors have also maintained an experimental study on shared disk Big Data analytics. They have compared HDFS (Apache Hadoop 1.0.2) which is the default file system of Apache Hadoop and Symantec Corporation's VERITAS Cluster File System (SFCFSHA 6.0) which is widely deployed by enterprises and organizations in banking, financial, telecom, aviation and various other sectors.

Mark Barlow in [11] covered a different angle of analytics tools in order to respond to real time events where data needs to be processed as it arrives rather than stored and retrieved later, as this process consumes more time to generate results. The author claims that having real time data gathered, these analytical tools which resides above the data layer are able to generate real time responses such as detecting fraud while someone is swiping their card, triggering an offer while a shopper is standing in a checkout line, or placing a relevant add when someone is browsing a webpage and take meaningful decisions. Barlow sketched out a practical real time Big Data analytics (RTBDA) roadmap that serves a variety of stakeholders by describing the five phases of real-time Big Data analytics framework: Data distillation, Model development, Validation and deployment, Real-time scoring, and Model refresh. He also discusses the four-layers of RTBDA technology stack proposed by David Smith: Data, Analytics, Integration, and Decision.

The era of Big Data and Big Data analytics has emerged during the last few years, and since then much research was conducted in this area but none of them have made the big picture of the whole process for collecting, processing, analyzing, and generating insights to add value from this Big Data. Since its most valuable and least researched, we will introduce a conceptual framework for identifying Big Data analytical solutions, to improve the understanding of Big Data analytics processes.

## **2. Background Definitions**

This section provides an overview of Big Data, Apache Hadoop and its core components: Hadoop Distributed File System HDFS, and YARN. Due to space constraint, some aspects are explained in a highly simplified manner. A detailed description of them can be found in [2][4][12][13][14][15][16][17][18][19][20][21].



### 2.1. *Big Data*

Big Data concept means a datasets which continues to grow so much that it becomes difficult to manage by using existing database management concepts and tools. The difficulty can be related to data capture, storage, search, sharing, analytics and visualization [12]. Big Data is defined by the leading IT industry research group Gartner [13] as: “Big Data are high-Volume, high-Velocity, and/or high-Variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization.” The Big Data spans across four dimensions: Volume, Velocity, Variety, and Veracity.

- Volume – The size of data is very large and is in terabytes and petabytes.
- Velocity –A conventional understanding of velocity typically considers how quickly the data is arriving and stored, and its associated rates of retrieval.
- Variety – It extends beyond the structured data, including unstructured data of all varieties: text, audio, video, posts, log files etc.[12]
- Veracity - Uncertainty of data and data trust worthiness [28]. The last V is introduced by IBM to cover the fact that data is keep changing so you can't trust the data for making decisions.

### 2.2. *Apache Hadoop*

Hadoop is the name that creator Doug Cutting's son gave to his stuffed toy elephants. He was looking for something that was easy to say and stands for nothing in particular[2].

Hadoop provides a distributed file system and a framework for the analysis and transformation of very large data sets using the MapReduce[14] paradigm. The important characteristic of Hadoop is the partitioning of data and computation across many (thousands) of hosts, and executing application computations in parallel close to their data. Hadoop is an Apache project; all components are available via the Apache open source license. Yahoo! has developed and contributed to 80% of the core of Hadoop [15].

Although Hadoop is best known for MapReduce and Hadoop Distributed File System (HDFS), the term is also used for a family of related projects that fall under the umbrella of infrastructure for distributed computing and large-scale data processing [2]. Briefly the core components for Hadoop ecosystem: HDFS (storage), and MapReduce 2.0 or YARN (resource managing and data processing). The other components are summarized at the end of this section. The use of components will depend on Hortonworks Data Platform (HDP) [16] as they use open source distribution powered by Apache Hadoop and they provide actual Apache-released versions of the components with all necessary bug fixes to make all the components interoperable in the production environment.

### 2.3. *Hadoop Distributed File System(HDFS)*

HDFS is the file system component of Hadoop designed for storing very large files with streaming data access patterns, running on clusters of commodity hardware [2]. HDFS stores file systems metadata and application data separately. As in other distributed file systems, such as, PVFS [17], Lustre [18] and GFS [19], HDFS stores metadata on a dedicated server, called the NameNode. Application data are stored on other servers called DataNodes. All servers are fully connected and communicate with each other using TCP-based protocols[20].

#### 2.4. YARN (*MapReduce 2.0*)

MapReduce was created by Google mainly to process big volume of unstructured data. MapReduce is a general execution engine that is ignorant of storage layouts and data schemas. The runtime system automatically parallels the computations across a large cluster of machines, handles failures and manages disk and network efficiency. The user only needs to provide a map function and a reduce function. The map function is applied to all input rows of the dataset and produces an intermediate output that is aggregated by the reduce function later to produce the final result [21].

In 2010, a group at Yahoo! began to design the next generation of MapReduce. The result was YARN shortened for Yet Another Resource Negotiator. YARN meets the scalability shortcomings of “classic” MapReduce”. YARN is more general than MapReduce, and in fact MapReduce is just one type of YARN application. The beauty of YARN’s design is that different YARN applications can co-exist on the same cluster, so a MapReduce application can run at the same time as an MPI (Message Passing Interface) application [2]. It performs the resource management function in Hadoop 2.0 and extends MapReduce capabilities by supporting non-MapReduce workloads associated with other programming models [16]. Which brings great benefits for manageability and cluster utilization [2].

The Hadoop Ecosystem is made up of three types of services – Core, Data, and Operational Components [2][16]:

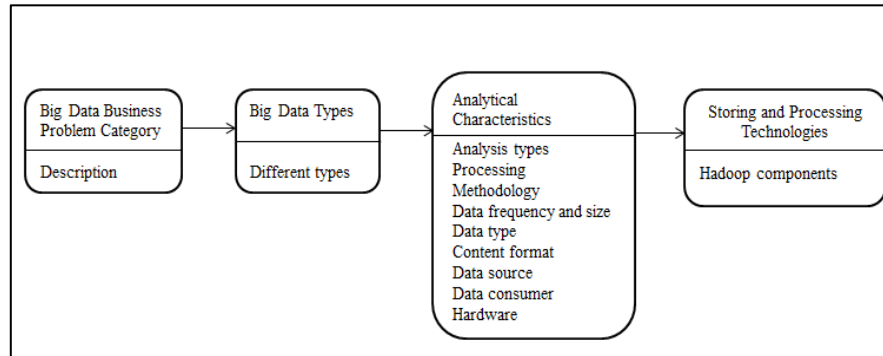
- The Core service component comprise of: HDFS, MapReduce, YARN
- The Data service component comprise of: Pig, Hive, HBase, Hcatalog, Strom, Mahout, Accumulo, Flume, and Scoop
- The Operational service component comprise of: Zookeeper, Ambari, Falcon, and Knox

### 3. Conceptual Framework for Identifying Big Data Analytics

A conceptual framework is useful for structuring contextual elements within complex settings. It is defined as a visual or written product, one that explains, either graphically or in narrative form. The main things to be studied are the key factors, concepts, or variables—and the presumed relationships among them. It is something that is constructed, not found. It incorporates pieces that are borrowed from elsewhere, but the structure, the overall coherence, is something that we build, not something that exists ready-made [23][24].

The conceptual framework defined in this research is aimed to structure the main elements of the Big Data analytics solutions. The basic concepts are Big Data business problem categories, Big Data types, analytical characteristics, and technologies. Using [22] we defined the Big Data business problem categories and their data types, which is important to be defined to make the process of identifying the appropriate analytic solution less complex and more ideal. For each Big Data category we analyze their characteristics: the format of the data that must be processed, the type of analysis to be applied, the processing techniques at work, and the data sources for the data that the target system is required to acquire, load, process, analyze, and store. The last concept is the technologies used to store and process Big Data. We chose Hadoop in our framework, because it is one of the most used platforms for Big Data Analytics in industry today [25]. It provides as you can see in the background section, various components to store, process Big Data. It been used as a base component for various vendor solutions, such as, HP [26], IBM [27].

The conceptual framework of Big Data processing is presented in Figure 2.



**Figure 2:** Conceptual Framework for Big Data Processing.

### 3.1. Research Analysis

#### 3.1.1. Initial Picture to Clarify the Technologies

Below in the Table 1 an initial picture that clarify the relationship between Big Data business problem categories, their data types, and suitable combination of Hadoop components. The required analytical tools are not specified because it is beyond the scope of this research.

**Table 1.** Summary of business problems Categories associated

Business Problem	Big Data Type	Hadoop Tools
Utilities: Predict power consumption	Machine-generated data	HDFS, YARN, MapReduce, Tez, Storm, Sqoop, Hbase, Hive, Accumulo, ZooKeeper.
Telecommunications: Customer churn analytics	Web and social data, Transaction data	HDFS, YARN, MapReduce, Tez, Storm, , HCatalog, Sqoop, Hbase, Hive, Accumulo, ZooKeeper.
Marketing: Sentiment analysis	Web and social data	HDFS, YARN, MapReduce, Tez, Hbase, Hive, Mahout, ZooKeeper.
Customer service: Call monitoring	Human-generated	HDFS, YARN, MapReduce, Hive, Pig, HCatalog, Mahout.
Retail: Personalized messaging based on facial recognition and social media	Web and social data, Biometrics	HDFS, YARN, MapReduce, Hbase, Hive, Pig, HCatalog, Mahout.
Retail and marketing: Mobile data and location-based targeting	Machine-generated data, Transaction data	HDFS, YARN, Tez, MapReduce, Pig, Hive, Mahout, ZooKeeper.
FSS, Healthcare: Fraud detection	Machine-generated data, Transaction data , Human-generated data	HDFS, YARN, MapReduce, Hbase, Storm, Flume, HCatalog, ZooKeeper, Accumulo.

#### 3.1.2. Conceptual framework for Big Data Analytic Solution

The research started with the curiosity to understand Big Data, and how it can be analyzed to extract value from it. This exploration journey faced different Big Data problems, because of the sheer volume, velocity, and variety of data the process of getting information and gaining insights become difficult. The starting point was on Big Data business problems' categories [22]. Then for each category their data types and their analytical characteristics were identified. After the categorizations started to

look for the technologies to retrieve, process, and the analyze Big Data. We explored various types of tools: open sourced and provided by vendors. We chose Hadoop in our framework, because it is one of the most used platforms for Big Data Analytics [23]. We used Hadoop as a base for Big Data solution, with all the provided features supporting to store and process Big Data.

Finally, the framework is extended with Big Data analytic and reporting tools. This framework is considering tools that analyze Big Data, and these tools are generally supported by vendors, such as Vertica offered by HP [26], InfoSphere and BigInsights offered by IBM [27], and many others. In the literature when they talked about Big Data analytics, they were more concerned about what tools are there to analyze Big Data. It can be appealing to just go out and buy Big Data analytics tool, thinking it will be the answer to business needs. But Big Data analytics technologies on its own aren't sufficient to handle the task. Well-planned processes needed to leverage the technologies that are essential to carry out an effective Big Data analytics. The well-planned analytical process have these essential concepts: business problem, data types, analytical characteristics, storing and processing, and analytical tools (see figure3).

We built our framework on the basis of these components to guide our thinking when one wants to analyze Big Data, and enable finding the best analytical solution for Big Data problem. We have not worked in detail on the mapping of vendor products to analyze characteristics, but we will consider that as future work.

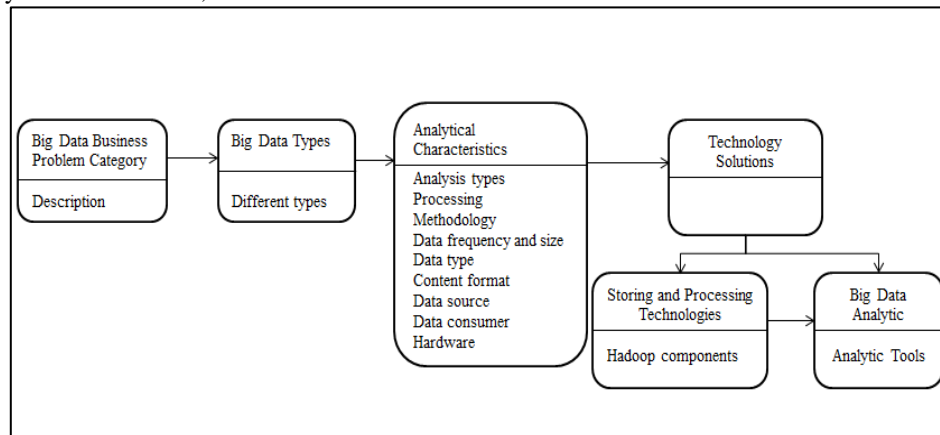


Figure 3: Conceptual Framework for Big Data Analytical Solution.

#### 4. Conclusion

In spite of the challenging analytical workload of Big Data, it is the key enabler for business evolution. Investigating the problem and adopting the right track to recognize patterns and reveal customers' insights from large volumes and variety of data gives new opportunities and resolves bottlenecks. This paper provided an overview of Big Data, described the open source framework, Hadoop, for managing and benefiting from these data, and produced a framework for mapping the right Big Data storage and processing tools to the corresponding business problem.

Future work will focus on the analytical tools which extract relations between pieces of data and to detect patterns and insights. Moreover, the application of testing this framework on a given case of business problem to evaluate its effectiveness will be covered as well. Thorough practical examination of the overall process and enhancements to the framework may prove necessary.

## References

- [1] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H., "Big Data: The next frontier for innovation, competition, and productivity", *McKinsey Global Institute*, May 2011.
- [2] White, Tom. *Hadoop: The Definitive Guide*, O'Reilly, Farnham, UK, 2012.
- [3] Dittrich, J., "Efficient Big Data Processing in Hadoop MapReduce", *VLDB Endowment*, vol. 5, no.12, pp. 2014-2015, August 2012.
- [4] The Apache Software Foundation, [Online] Available at: <http://hadoop.apache.org/>.
- [5] Anchan, P., "Top 5 Most Important Companies in Big Data", Oct 25, 2013, [Online] available at: <http://www.cloudcomputingpath.com/top-5-most-important-companies-in-big-data/>
- [6] HP, [White paper] "HP Vertica Enterprise Edition software", July 2012.
- [7] Katal, A., Wazid, M., Goudar, R.H., "Big Data: Issues, challenges, tools and Good practices" *IEEE Contemporary Computing*, pp. 404-409, 2013.
- [8] Chandrasekhar, U., Redday, A., Rath, R., "A Comparative Study of Enterprise and open source Big Data analytical tools", *IEEE Information & Communication Technologies*, pp. 372-377, 2013.
- [9] Azemovic, J., Music, D., "Comparative analysis of efficient methods for storing unstructured data into database with accent on performance" , *IEEE Education Technology and Computer*, pp. V1-403 - V1-407, 2010.
- [10] Mukherjee, A., Datta, J., Jorapur, R., Singhvi, R., Haloi, S., Akram, W., "Shared Disk Big Data Analytics with Apache Hadoop", *IEEE High Performance Computing*, pp. 1-6, 2012.
- [11] Barlow, M., "Real-Time Big Data Analytics: Emerging Architecture", *O'Reilly Media*, Feb. 2013.
- [12] Singh, S., Singh, N., "Big Data analytics," *Communication, Information & Computing Technology (ICCICT), 2012 International Conference*, pp.1,4, 19-20 Oct. 2012
- [13] C. Regina, M. Beyer, M. Adrian, T. Friedman, D. Logan, F. Buytendijk, M. Pezzini, R. Edjlali, A. White, and D. Laney, "Top 10 Technology Trends Impacting Information Infrastructure, 2013". [Online]. Available: <http://my.gartner.com/portal/server.pt?open=512&objID=256&mode=2&PageID=2350940&resId=2340315&ref=QuickSearch&stkw=Top+10+Technology+Trends+Impacting+Information+Infrastructure%2C+2013>, 19 Feb. 2013
- [14] J. Dean, S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *In Proc. of the 6th Symposium on Operating Systems Design and Implementation*, San Francisco CA, Dec. 2004.
- [15] Kurazumi, S.; Tsumura, T.; Saito, S.; Matsuo, H., "Dynamic Processing Slots Scheduling for I/O Intensive Jobs of Hadoop MapReduce," *Networking and Computing (ICNC), 2012 Third International Conference*, pp.288,292, 5-7 Dec. 2012
- [16] Hortonworks, [Online] Available at: <http://hortonworks.com/>
- [17] P. H. Carns, W. B. Ligon III, R. B. Ross, and R. Thakur. "PVFS: A parallel file system for Linux clusters," *in Proc. of 4th Annual Linux Showcase and Conference*, 2000, pp. 317-327.
- [18] Lustre File System.[Online] available at: <http://www.lustre.org>.
- [19] M. K. McKusick, S. Quinlan. "GFS: Evolution on Fast-forward," *ACM Queue*, vol. 7, no. 7, New York, NY. August 2009.
- [20] Shvachko, K.; Hairong Kuang; Radia, S.; Chansler, R., "The Hadoop Distributed File System," *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium* , pp.1,10, 3-7 May 2010
- [21] Xiongpai Qin; Huiju Wang; Furong Li; Baoyao Zhou; Yu Cao; Cuiping Li; Hong Chen; Xuan Zhou; Xiaoyong Du; Shan Wang, "Beyond Simple Integration of RDBMS and MapReduce -- Paving the Way toward a Unified System for Big Data Analytics: Vision and Progress," *Cloud and Green Computing (CGC), 2012 Second International Conference* , pp.716,725, 1-3 Nov. 2012.
- [22] D. Mysore, S. Khupat, S. Jain, "Big Data architecture and patterns, Part1: Introduction to Big Data classification and architecture", IBM Corp., 17 Sep. 2013. [Online] Available: <http://www.ibm.com/developerworks/library/bd-archpatterns1/>
- [23] Catherine D. Ennis, "Conceptual Frameworks as a Foundation for the Study of Operational Curriculum", *Journal of Curriculum and Supervision*, vol. 2, no. 1, pp. 25-39, Fall 1986.
- [24] Joseph A. M., *Qualitative Research Design: An Interactive Approach*, 3<sup>rd</sup> Edition , Sagepub, 2012.
- [25] Weiyi Shang; Zhen Ming Jiang; Hemmati, H.; Adams, B.; Hassan, A.E.; Martin, P., "Assisting developers of Big Data Analytics Applications when deploying on Hadoop clouds," *Software Engineering (ICSE), 2013 35th International Conference* , pp.402,411, 18-26 May 2013.
- [26] Gareth M., "Big Data Big Opportunities", *HP Corp.*, Sept. 2013.
- [27] Chris E., Dirk D., Tom D., George L., Paul Z., "Understanding Big Data, Analytics for Enterprise Class Hadoop and Streaming Data", IBM Cor., 2012.
- [28] "The Big Data and Analytic Hub", IBM, [Online] Available at: <http://www.ibmbigdatahub.com/infographic/four-vs-big-data>

# Flexible Information Integration with Local Dominance

Scott BRITELL<sup>a</sup>, Lois M.L. DELCAMBRE<sup>a</sup>, and Paolo ATZENI<sup>b</sup>

<sup>a</sup> *Portland State University, PO Box 751, Portland, OR, 97207-0751, USA*

<sup>b</sup> *Università Roma Tre, Via della Vasca Navale 79, 00146 Roma, Italy  
{britell,lmd}@cs.pdx.edu, atzeni@dia.uniroma3.it*

**Abstract.** Domain-specific web applications often need to integrate information from schematically heterogeneous sources that share some but not all semantic similarities. These applications often include application widgets—where each widget may address a (potentially small) subset of the local schema. We seek to provide flexible integration where each widget may use its own “global” schema and use its own mapping to each local schema. Note that different mappings, even to the same local schema, may be quite different or even contradictory. Traditional information integration is too rigid to meet these requirements. Here, we define a new integration model that introduces a metamodel of small domain-specific schema fragments—called domain structures—that can be mapped to local schemas. We show how generic, polymorphic widgets can be created by writing queries against domain structures using an extended relational algebra that includes a local type operator to propagate local semantics to the domain structures.

**Keywords.** Information integration, query languages, Web application, conceptual models

## 1. Introduction

Modern web development environments allow non-expert users to create web sites and applications with underlying schemas and data storage capabilities. Our goal is to make it easy for these users to integrate their information from schematically heterogeneous sources that share semantic similarities; but, where the individual local schemas have important semantics. And, where these local semantics (e.g., local attribute, entity, or relationship type names), whether integrated or not, can be useful for the end-user to see. For example, a sports application may integrate information about boxing matches and baseball games for use by a widget that shows nearby competitions to a user. But simply integrating the two types of competitions to a generic global schema would lose the semantic differences between the two such as the fact that a boxing match is between two individual people whereas a baseball game is between two teams. In the same application, we may want another widget that allows the user to browse all athletes, no matter which sport, requiring that the boxers be integrated with baseball players instead of entire teams. It would also be useful to be able to compose integrations from the athlete widget and the competition widget to show relevant information about boxers in a competition.

Information integration has been long studied in the fields of databases and the semantic web but traditional techniques lack the flexibility we would like. In traditional in-

formation integration there is usually a single rigid global schema, where only the semantics of that global schema are available to query writers. Traditionally, global schemas are generally not composable. Another issue with traditional information integration is the need for a database specialist to create mappings and transformations between global and local schemas. This problem has been one of the stumbling blocks to the adoption of the semantic web [1]. Recently search engines have incentivized integration by providing widgets to users who semantically annotate their databases with known global schemas, but users are required to understand complex technologies and these widgets are still limited to the global semantics. We want to solve this problem by enabling mappings based on simple correspondences that can be created by drawing a line from global schema fragments to local schemas and by providing a mechanism to create generic global widgets that can show local semantics. Our goal is for users to get as little or as much functionality by mapping as little or as much as they want.

Here we present our model for information integration with local dominance that is built upon small global schema fragments (similar to data modeling patterns) called *domain structures*. We show how domain structures are mapped to local schemas, and how, using these mappings, we can write queries at the domain level that return results from mapped local schemas with local semantics added. We formally define our model, mappings, and extend the relational algebra for our structures and to incorporate the local type semantics. We show how domain structures are used in generic polymorphic widgets and how domain structures may be composed for more complex tasks. We note that users can use our generic widgets even when there is only one local schema (without integration).

In general, our work has the following goals for information integration:

- **simplicity** - Mappings should be simple; we envision end-users who understand the application domain to be able to describe mappings.
- **multiplicity** - We envision the use of many domain structures, with many different (perhaps even contradictory) mappings, in a single web application.
- **composability** - Domain structures should be easily composable so that users need only map the domain structures of interest while other users may make use of those mappings in more complex composed structures.
- **flexibility** - Our mappings and other constructs should be largely unconstrained; we want to permit the construction of various, polymorphic widgets.
- **tolerance** - We expect our mapping and querying infrastructure to work even for what appear to be unusual mappings (based on our flexibility).
- **local dominance** - A query writer should be able to introduce local schema information into a widget—for any construct in a domain structure.
- **genericity/polymorphism** - Widgets should be written against domain structures and thus work generically on a broad range of local schemas.
- **immediacy** - Through the use of widgets, an end-user should be able to see the effects of their mappings directly. This enables pay-as-you-go integration.

## 2. Motivating Example

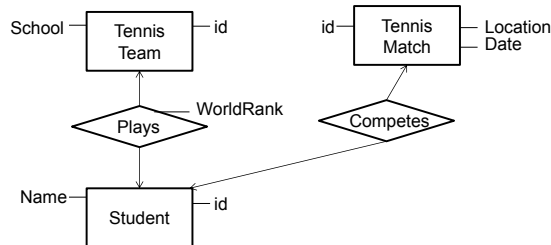
As a motivating example, we introduce a fictional sports application that integrates information about sports teams and their matches. Imagine we have two database instances

described by the schemas in Figures 1 and 2. In the *Tennis* schema (Figure 1) we describe university tennis teams (*TennisTeam*) and the students who play on those teams (*Student* and *Plays*). We also describe tennis matches between different students. In the *Football* schema (Figure 2) we describe football teams (*FootballTeam*) and the people who manage or play for the team (*Person*, *Manages*, and *PlaysFor*). Football games are then shown as between different football teams.

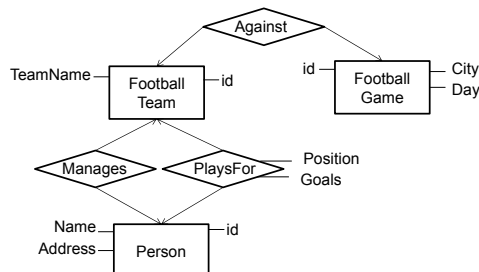
We show two example widgets that we would like to create generically to work over such sports databases. In Figure 3 we see a team browser across different sports. This widget combines information about students playing for tennis teams as well as people playing for football teams. This widget combines semantically similar concepts from each schema (people playing for teams) but brings across the local semantic information of which type of team the person plays for, and, in the case of the football schema, the position the person plays.

Figure 4 shows a widget that lists competitions in the site. Here we see tennis matches between different individual people and football games between different teams. This example shows how there are similar semantics between football teams and students in the context of a competition that weren't applicable to the team widget. This widget also shows how we can compose the output of the first widget, in the case of football teams, to provide more information.

These two widgets show how 1) the local semantics are important—without knowing the difference between football teams vs. students and football games vs. tennis matches we wouldn't be able to tell the difference between the “Giants” and “Bob Smith” in the competition widget—and 2) we can benefit from small, composable global schema fragments to produce widgets like the competition widget.



**Figure 1.** A simple tennis schema.



**Figure 2.** A simple football schema.





Figure 3. A generic team widget.

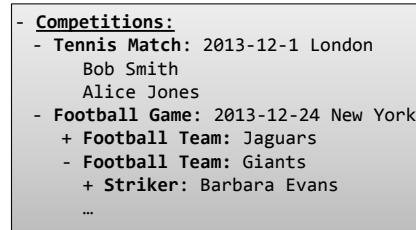


Figure 4. A generic competition widget.

### 3. Domain Structures for Locally-Dominant Integration

Figure 5 shows an overview of the architecture of our system. At the bottom of the figure are the local schemas for which we wish to provide generic functionality and widgets. We define small global schema fragments called domain structures, shown in the middle of the figure. Mappings are defined between the domain structures and the local schemas consisting of simple correspondences. Using these mappings we define how a domain structure is applied to local schemas to provide an integrated query environment. We use a relational query language to query a relational form of our domain structures that returns integrated results from the various local schemas. We extend the relational query language by defining a local type operator,  $\tau$ , to bring local semantics to the global level. In this section, we begin with the background of the Entity-Relationship (ER) model in the first subsection and then present our new contributions in the following subsections.

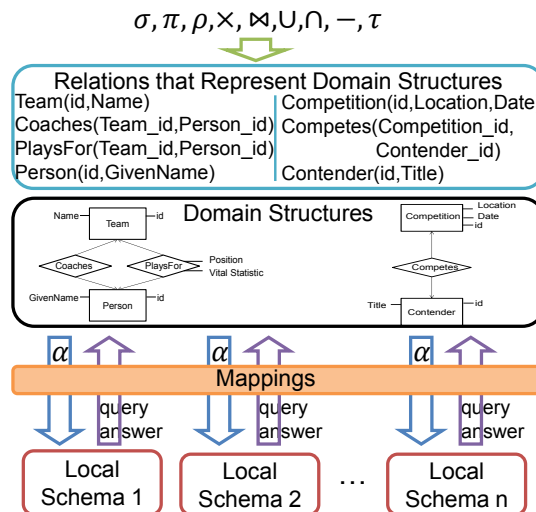


Figure 5. Architecture Figure.

### 3.1. ER Background

We begin by first recalling the definitions of attributes, entities, and relationships from the ER model as shown in Thalheim [2]. We use this ER description to describe the local schemas and we will extend these basic definitions to define our domain structures.

A **data scheme**  $DD = (U, \underline{D}, dom)$  is a finite set  $U$  of **simple attributes** (i.e., attribute names)  $\{A_1, A_2, \dots\}$ , a set  $\underline{D} = \{D_1, D_2, \dots\}$  of **domains**, and a **domain function**  $dom : U \rightarrow \underline{D}$  which associates an attribute with its domain.

An **entity type**  $E = (attr(E), id(E))$  is a set of attributes from  $U$  given by the function  $attr(E)$  and an id for the entity given by the function  $id(E)$ , where  $E$  is the name of the entity.

A **relationship type**  $R = (ent(R), attr(R))$  is a sequence of entity types given by the function  $ent(R)$  and a set of attributes from  $U$  given by the function  $attr(R)$ , where  $R$  is the name of the relationship.

We also define  $ctype(LT)$  to be a function that given a local ER type  $LT$  returns the construct type, i.e., “attribute”, “entity”, or “relationship” depending on the type of  $LT$ .

### 3.2. Domain Structures

A domain structure is a schema fragment that is analogous to a global schema in traditional integration. We define three **domain structure types** in this section. For simplicity, in the paper we assume that domain structure type names are unique.

A **domain attribute** is a name  $DA$ . We do not associate a domain with the domain attribute; domain information is available from a local schema whenever there is a mapping from the domain attribute to local attributes. In Figure 6 there are six domain attributes  $Team\_id$ ,  $Name$ ,  $Position$ ,  $VitalStatistic$ ,  $Person\_id$ , and  $GivenName$ . In Figure 7 there are 5 domain attributes  $Location$ ,  $Date$ ,  $Competition\_id$ ,  $Contender\_id$ , and  $Title$ .

A **domain entity**  $DE = (Dattr(DE), id(E))$  is a set of domain attributes given by the function  $Dattr(DE)$  and an id attribute given by the function  $id(E)$ , where  $DE$  is the name of the domain entity. In Figures 6 and 7 there are four domain entities:

$$Team = (\{Name\}, Team\_id)$$

$$Person = (\{GivenName\}, Person\_id)$$

$$Competition = (\{Location, Date\}, Competition\_id)$$

$$Contender = (\{Title\}, Contender\_id)$$

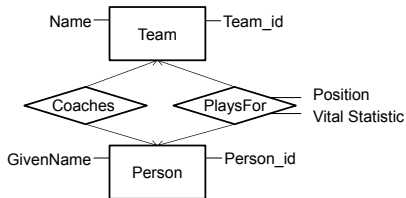


Figure 6. The *Team* domain structure.

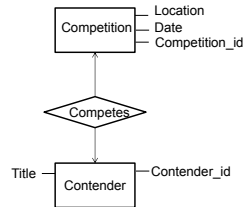


Figure 7. The *Competition* domain structure.

A **domain relationship**  $DR = (Dent(DR), Dattr(DR))$  is a sequence of domain entity types given by the function  $Dent(DR)$  and a set of domain attributes given by the function  $Dattr(DR)$ , where  $DR$  is the name of the domain relationship. For example, in Figure 6 we define the *Coaches* and *PlaysFor* domain relationships and in Figure 7 we define the *Competes* domain relationship.

$$Coaches = ((Person, Team), \{\})$$

$$PlaysFor = ((Person, Team), \{Position, VitalStatistic\})$$

$$Competes = ((Contender, Competition), \{\})$$

A **domain structure**  $DS$  is defined as a set of domain structure types. The team and competition domain structures are defined as follows:

$$TeamDS = \{Coaches, PlaysFor, Person, Team\}$$

$$CompetitionDS = \{Competes, Contender, Competition\}$$

Domain structures may be composed through the union operation, creating new domain structures. For example, we can compose  $TeamDS$  and  $CompetitionDS$  to create a new domain structure.

$$\begin{aligned} TeamDS + CompetitionDS &= TeamDS \cup CompetitionDS \\ &= \{Coaches, PlaysFor, Person, Team, Competes, \\ &\quad Contender, Competition\} \end{aligned}$$

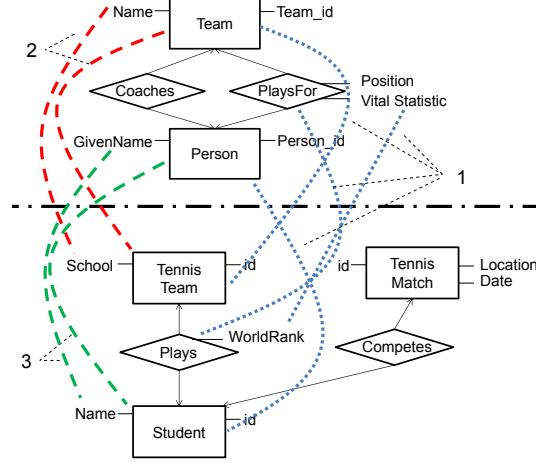
Note that we do not constrain the domain structure types that can be part of the domain structure, in keeping with our goals of simplicity, flexibility and composability. Having small unconstrained domain structures allows them to be composed in different ways for different purposes.

### 3.3. Mappings

In order to access local schemas through domain structures, we define mappings from the domain structures to the local schemas. We map from domain entities and domain relationships to local schemas; domain attributes in those domain structure types are mapped as we map the domain structure type. Our mappings are sets of simple correspondences. So in order to define mappings, we first define correspondences.

A **correspondence**  $C = (DT(C), LT(C), id(C))$  is a domain structure type given by the function  $DT(C)$ , a local type given by the function  $LT(C)$ , and the id, of the mapping to which the correspondence belongs, given by the function  $id(C)$ . A correspondence may be from any domain structure type (domain attribute, domain entity, or domain relationship) to the corresponding local type (attribute, entity, or relationship).

A **mapping**  $M = (DT(M), corr(M), id(M))$  for a domain entity or domain relationship is a domain structure type given by the function  $DT(M)$ , a set of correspondences given by the function  $corr(M)$ , and an identifier for the mapping given by the



**Figure 8.** Schema-Structure mapping of the *Team* domain structure (above) to the tennis schema (below). The schema-structure mapping consists of three mappings: 1) (dotted/blue lines) the *PlaysFor* domain relationship is mapped to the *Plays* local relationship, 2) (dashed/red lines) the *Team* domain entity is mapped to the *TennisTeam* local entity, and 3) (solid/green lines) the *Person* domain entity is mapped to the *Student* local entity.

function  $id(M)$ . We do not constrain the correspondences that comprise a mapping to be consistent with our goals of: simplicity, end-user mapping, tolerance, and immediacy. For example, if a domain entity has three domain attributes and an end-user maps only one of those our system will only return the one mapped attribute. In Figure 8 there are three mappings:

$$M_1 = (PlaysFor, \{(PlaysFor, Plays, 1), (VitalStatistic, WorldRank, 1), (Team, TennisTeam, 1), (Person, Student, 1)\}, 1)$$

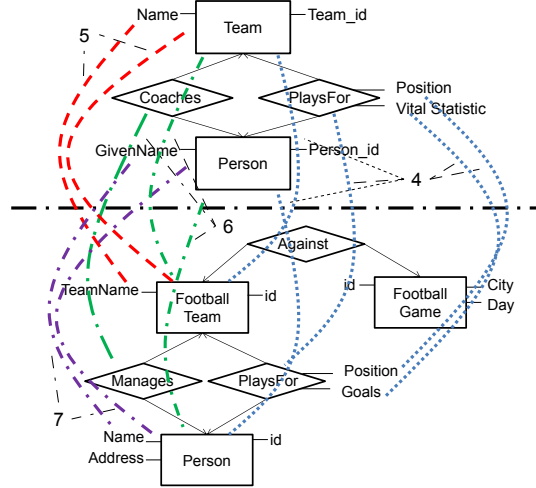
This mapping corresponds to the blue/dotted lines in Figure 8. This mapping represents mapping the *PlaysFor* domain relationship and its *VitalStatistic* attribute to the *Plays* relationship and its *WorldRank* attribute in the Tennis schema.

$$M_2 = (Team, \{(Team, TennisTeam, 2), (Name, School, 2)\}, 2)$$

This mapping corresponds to the red/dashed lines in Figure 8. This mapping represents mapping the *Team* domain entity with its *Name* attribute mapped to the *TennisTeam* entity and its *School* attribute in the Tennis schema.

$$M_3 = (Person, \{(Person, Student, 3), (GivenName, Name, 3)\}, 3)$$

This mapping corresponds to the green/solid lines in Figure 8 mapping the *Person* domain entity and its attribute to the *Student* local entity.



**Figure 9.** Schema-Structure mapping of the *Team* domain structure to the football schema. The schema-structure mapping consists of four mappings: 4) (dotted/blue lines) the *PlaysFor* domain relationship is mapped to the *PlaysFor* local relationship, 5) (dashed/red lines) the *Team* domain entity is mapped to the *FootballTeam* local entity, 6) (solid/green lines) the *Coaches* domain relationship is mapped to the *Manages* local relationship, and 7) (dashed-dotted/purple lines) the *Person* domain relationship is mapped to the *Person* local relationship.

Mappings  $M_4 \dots M_{13}$  are shown in Figures 9, 10, and 11 and are represented similarly to the above mappings. We do not elaborate them here.

Note, we do not place any constraints on the number of times a domain structure may be mapped to a local schema. If a user creates multiple identical mappings, data from the mapped local schema will be duplicated in queries over that domain structure.

Next, we define the collection of mappings from a domain structure to a local schema as a **schema-structure mapping**

$$SSM = (DS(SSM), S(SSM), mset(SSM), id(SSM))$$

where  $DS(SSM)$  is a function returning a domain structure name,  $S(SSM)$  is a function returning a local schema name,  $mset(SSM)$  is a function returning a set of mappings, and  $id(SSM)$  is an function returning the identifier for the schema-structure mapping.

The schema-structure mappings shown in Figures 8, 9, 10, and 11 are shown below. Note: we have numbered the schema-structure mappings here by the figures to which they correspond in order to facilitate readability.

$$SSM_8 = (TeamDS, Tennis, \{M_1, M_2, M_3\}, 8)$$

$$SSM_9 = (TeamDS, Football, \{M_4, M_5, M_6, M_7\}, 9)$$

$$SSM_{10} = (CompetitionDS, Tennis, \{M_8, M_9, M_{10}\}, 10)$$

$$SSM_{11} = (CompetitionDS, Football, \{M_{11}, M_{12}, M_{13}\}, 11)$$

### 3.4. Query Language

In this section we discuss our extensions to the standard relational algebra. For this section we treat our local schemas and domain structures as relations where each entity  $E$  is represented as a relation consisting of its attributes and each relationship  $R$  is represented as a relation consisting of the id attributes of its entity components followed by its attributes. Examples of this for the *Tennis* and *Football* schemas can be seen in Figures 12 and 13, respectively.

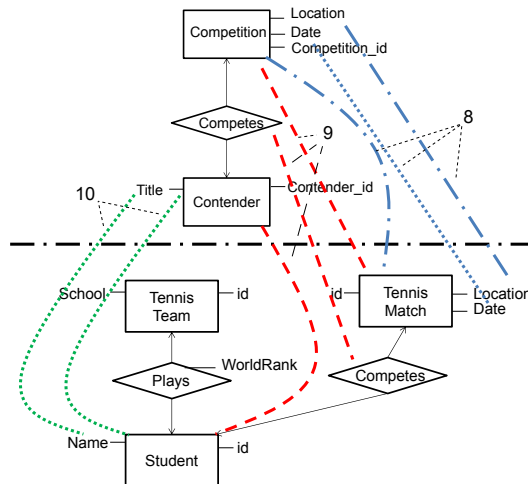
As in any query language we must first define what queries will address. Traditionally, this would be an instance of the integrated database; here we analogously define a domain structure instance against which queries may be written.

A **domain structure instance**  $DSi = (DS, SSM(DSi))$  is a domain structure  $DS$  and a function returning a set of schema-structure mappings  $SSM(DSi)$ .

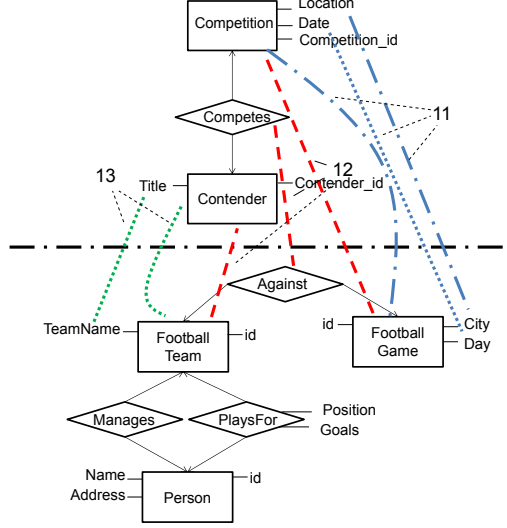
We allow the set returned by  $SSM(DSi)$  to contain any number of schema-structure mappings since a query writer may want to use domain structures to benefit a single local schema (e.g. where there is just one schema-structure mapping) but a different query writer may want to use multiple local schemas in their queries (where there would be multiple schema-structure mappings).

A query over a traditional database begins by choosing which relations to query and implicitly includes a *tablescan* operation to allow the rest of the query operators to affect the result of that scan. We analogously define the apply operation that applies a domain structure type to its mapped local schemas. We define the apply operator for domain entity and domain relationship types.

We first define several constructs in order to simplify the definition of the apply operator. Given a domain structure type  $DT$  and a domain structure instance  $DSi$ , let



**Figure 10.** Schema-Structure mapping of the *Competition* domain structure to the tennis schema. The schema-structure mapping consists of three mappings: 8) (solid/blue lines) the *Competition* domain entity is mapped to the *TennisMatch* local entity, 9) (dashed/red lines) the *Competes* domain relationship is mapped to the *Competes* local relationship, and 10) (dotted/green lines) the *Contender* domain entity is mapped to the *Student* local entity.



**Figure 11.** Schema-Structure mapping of the *Competition* domain structure to the football schema. The schema-structure mapping consists of three mappings: 11) (solid/blue lines) the *Competition* domain entity is mapped to the *FootballGame* local entity, 12) (dashed/red lines) the *Competes* domain relationship is mapped to the *Against* local relationship, and 13) (dotted/green lines) the *Contender* domain entity is mapped to the *FootballTeam* local entity.

$$\begin{aligned}
corr &= \{C | SSM \in SSM(DSi) \wedge M \in mset(SSM) \\
&\quad \wedge C \in corr(M)\} \\
LTi(DT) &= \{(LT, mid) | C \in corr \wedge LT = LT(C) \wedge mid = id(C) \\
&\quad \wedge DT = DT(C)\} \\
ADattr(mid) &= \{(A, DA, mid) | C \in corr \wedge A = LT(C) \wedge mid = id(C) \\
&\quad \wedge DA = DT(C) \wedge ctype(A) = \text{"attribute"}\} \\
&\cup \{(id, id(DT), mid) | C \in corr \wedge DT = LT(C) \wedge mid = id(C) \\
&\quad \wedge ctype(LT) = \text{"entity"}\}
\end{aligned}$$

Here, we first define sets for the correspondences, local types, and domain and local attributes needed for the apply operation. We begin by getting all of the correspondences of the schema-structure mappings of  $DSi$  in  $corr$ . From these correspondences we can then determine the local types that the domain structure has been mapped to and their mapping ids,  $LTi$ . From the mapping ids we can then also determine all of the attributes that are in those mappings and their corresponding domain structure types,  $ADattr(mid)$ . We also add the domain entity ids and the local entity ids for every domain entity that is mapped to a local entity in the second component of the union. Since we assume that all local entities have the id attribute we do not explicitly need to map this attribute. The

domain entity to local entity mapping is sufficient. Then **applying** a domain structure type  $DT$  given a domain structure instance  $DSi$  is

$$\alpha(DT) = \bigcup_{(LT, mid) \in LTi(DT)} (\rho_{\{A \rightarrow DA \mid (A, DA, mid) \in AAttr(mid)\}} \left( \prod_{\{A \mid (A, DA, mid) \in AAttr(mid)\}} (LT) \right) \times (mid \rightarrow_{DT} mid)$$

For each local type that the domain structure type is mapped to, we first project all mapped attributes, rename them to their domain structure counterparts and then cross product that result with the mapping id that the data comes from. The notation  $\rightarrow_{DT}$  here means that we are naming this attribute as the domain structure type name  $DT$  concatenated with the string “\_mid”. Lastly we union the results from all the local schemas mapped in the schema-structure mappings of  $DSi$ .

Once domain structures have been applied using  $\alpha$ , standard relational algebra operations—union, difference, projection, product, join, selection, rename—can be used with the result without any change in definition.

In order to bring semantics from the local schemas to the domain structure instance we next define the **local type** operation  $\tau_{DT}(\chi)$ , which allows local type names to be introduced as new attributes into an existing relation  $\chi$  (whether as the result of other relational operations or from the apply operator). The type name can be an attribute, entity, or relationship name. We define  $\tau_{DT}(\chi)$  for a domain structure type  $DT$  and a domain structure instance  $DSi$  as follows:

$$\begin{aligned} \tau_{DT}(\chi) = \chi \bowtie_{DT\_mid=DT\_mid} & \left( \{ (LT \rightarrow_{DT\_type}, mid \rightarrow_{DT\_mid}) \right. \\ & \mid SSM \in SSM(DSi) \wedge M \in mset(SSM) \\ & \wedge C \in corr(M) \wedge LT = LT(C) \wedge mid = id(C) \\ & \left. \wedge DT = DT(C) \right) \end{aligned}$$

Looking at the set notation on the right side of the join operator we see a set of tuples of local type names and mapping ids where the mapping id comes from the domain structure instance  $DSi$  and the mappings come from the schema-structure mappings in  $DSi$ . The local types are then retrieved from the correspondences with that mapping id. We use the “ $\rightarrow$ ” notation to signify that we are setting the attribute name of the tuple to be the domain structure type name  $DT$  given in  $\tau$  concatenated with the string “\_type” or “\_mid” respectively. Lastly, we join these tuples with the given relation  $\chi$  on the mapping id.

Since all parts of the domain structure and the mapping id attributes can be accessed in the query answer there is the possibility that any may be projected out. In the case that the mapping ids are dropped from a query answer, the type operation on the resulting tuple will generate a null type.



TennisTeam	
id	School
1	Portland State University
2	University of Oregon

Plays		
TennisTeam_id	Student_id	WorldRank
1	1	1200
2	2	412

Student	
id	Name
1	Bob Smith
2	Alice Jones

Competes	
Student_id	TennisMatch_id
1	1
2	1

TennisMatch		
id	Location	Date
1	London	2013-12-1

Figure 12. Tennis schema as relations with sample data.

#### 4. Widgets

In this section we describe how data displayed in the widgets in Figures 3 and 4 can be produced by writing queries against elements of domain structures using our extended algebra. We first begin by looking at the relational instances of the tennis and football schemas with sample data to populate the two widgets.

Figure 12 shows the tennis instance where the *Student*, *TennisTeam*, and *TennisMatch* relations have all of the attributes of the corresponding entities in the local schema shown in Figure 1, and the *Plays* and *Competes* relationships have the id attributes of the related entities.

In Figure 13 the *FootballGame*, *FootballTeam*, and *Person* relations have all of the attributes of the corresponding entities and the *Against*, *Manages*, and *PlaysFor* relations have the corresponding ids of the entities contained in the relationships of the local schema shown in Figure 2.

We next show the query to create the team widget in Figure 3. We query the domain structure instance  $DSi = (Team, \{SSM_8, SSM_9\})$  comprising the *Team* domain structure and the schema-structure mappings shown in Figures 8 and 9. Our query involves joins and our  $DSi$  involves two schema-structure mappings. In this case, we only want to join data that exists in the same local instance so we define the following function:

$$\begin{aligned}
 S(DT_1, DT_2) = \{ & (mid_1, mid_2) \mid SSM \in SSM(DSi) \wedge \\
 & M \in mset(SSM) \wedge DT_1 \in DT(M) \wedge DT_2 \in DT(M) \wedge \\
 & mid_1, mid_2 \in mset(M)\}
 \end{aligned}$$

$S(DT_1, DT_2)$  is defined within a domain structure instance  $DSi$  and, given two domain structure types  $DT_1$  and  $DT_2$ , returns all pairs of mapping ids that are part of

Against	
FootballTeam_id	FootballGame_id
1	1
2	1

FootballGame		
id	City	Day
1	New York	2013-12-24

FootballTeam	
id	School
1	Jaguars
2	Giants

Manages	
FootballTeam_id	Person_id
2	2

PlaysFor			
FootballTeam_id	Person_id	Position	Goals
2	1	Striker	10

Person		
id	Name	Address
1	Barbara Evans	123 Main St
2	George Johnson	42 N. Elm St

Figure 13. Football schema as relations with sample data.

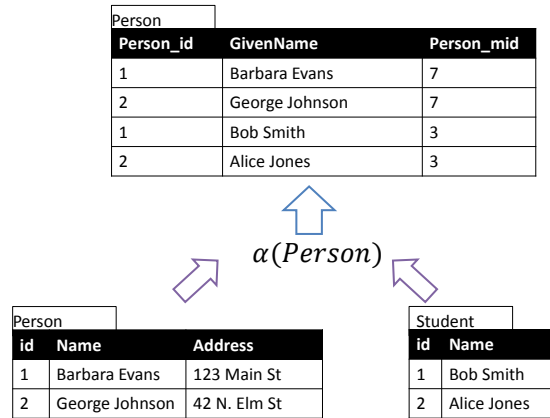
the same schema-structure mapping in  $DSi$ .<sup>1</sup> For example, given the schema-structure mappings shown in Figures 8 and 9,  $S(Team, PlaysFor) = \{(2, 1), (5, 4)\}$  and  $S(PlaysFor, Person) = \{(1, 3), (4, 7)\}$ .

$TeamQuery =$

$$\begin{aligned}
& \pi_{Name, Position, VitalStatistic, GivenName, Team\_type, Person\_type, Team\_mid} \left( (\tau_{Team}(\alpha(Team))) \right. \\
& \quad \bowtie_{Team\_id=Team\_id \wedge (Team\_mid, PlaysFor\_mid) \in S(Team, PlaysFor)} (\alpha(PlaysFor)) \\
& \quad \left. \bowtie_{Person\_id=Person\_id \wedge (PlaysFor\_mid, Person\_mid) \in S(PlaysFor, Person)} (\tau_{Person}(\alpha(Person))) \right)
\end{aligned}$$

Reading the query from right to left, we first apply the *Person* domain entity (Figure 14 shows an example of this apply to the Tennis and Football instances) and add type meta-data for the entity type. We then join that with the application of the *PlaysFor* domain relationship—this application is shown in Figure 15. In this case we join on the id of the *Person* domain entity as well as joining on correspondences from the same schema-structure mapping—this ensures that we do not mix data from different local schemas or from different mappings to the same local schema. We then join this result with the application of the *Team* domain entity (Figure 16 with the added entity type). Once again we join on the id of the domain entity and domain relationship as well as the correspondences from the same schema-structure mapping. Lastly, we project the *Name*, *Postition*, *VitalStatistic*, *GivenName*, *Team\_type*, *Person\_type*, and *Team\_mid* from the resulting joins.

<sup>1</sup>In this example, we want to explicitly separate the data from the different schema-structure mappings. Other queries that join across different schema-structure mappings are also possible and can be of use in scenarios such as entity resolution across different local schemas.

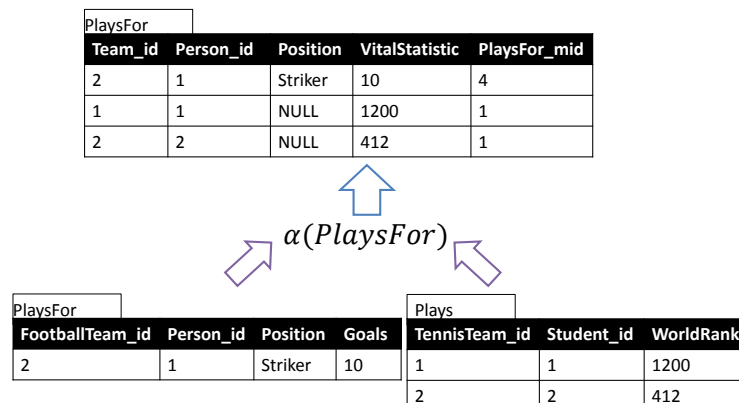


**Figure 14.** Applying the *Person* domain entity over tennis and football schemas using mappings from Figures 8 and 9.

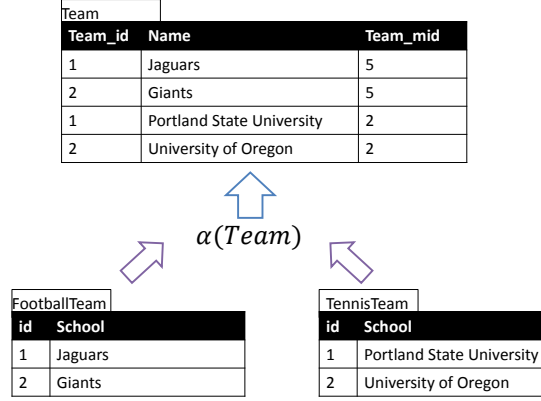
The resulting relation of the above query is shown in Figure 17. We can see how the widget shown in Figure 3 can be programmatically generated by nesting the data in this relation on first the *Name* attribute, then on the *GivenName* attribute.

In a similar fashion we write a query for the competition widget in Figure 4. For this widget we use a more complex domain structure instance  $DS_i = (Team + Competition, \{SSM_8, SSM_9, SSM_{10}, SSM_{11}\})$  that consists of the domain structure *Team + Competition* which is the composition of the *Team* and *Competition* domain structures defined in Section 3.2 and using the schema-structure mappings shown in Figures 8, 9, 10, and 11.

Since we are querying over differently mapped domain structures, we define the following function  $MLS(DT_1, DT_2)$  that given two domain structure types returns all mapping id pairs such that each mapping id belongs to a schema-structure mapping in the domain structure instance, and each of those schema-structure mappings is to the same local schema.



**Figure 15.** Applying the *PlaysFor* domain relationship over tennis and football schemas using mappings from Figures 8 and 9.



**Figure 16.** Applying the *Team* domain entity over tennis and football schemas using mappings from Figures 8 and 9.

TeamQuery						
Name	Position	Vital Statistic	GivenName	Team_type	Person _type	Team _mid
Giants	Striker	10	Barbara Evans	Football Team	Person	5
Portland State University	NULL	1200	Bob Smith	TennisTeam	Student	2
University of Oregon	NULL	412	Alice Jones	TennisTeam	Student	2

**Figure 17.** The relational output of the *TeamQuery* query.

$$\begin{aligned}
 MLS(DT_1, DT_2) = \{ & (mid_1, mid_2) \mid SSM_1 \in SSM(DSi) \wedge LS_1 \in LS(SSM) \wedge \\
 & SSM_2 \in SSM(DSi) \wedge LS_2 \in LS(SSM) \wedge LS_1 = LS_2 \wedge \\
 & mid_1 \in id(M_1) \wedge M_1 \in mset(SSM_1) \wedge \\
 & M_2 \in mset(SSM_2) \wedge mid_2 \in id(M_2) \}
 \end{aligned}$$

For example, given the schema-structure mappings shown in Figures 8,9,10, and 11,  $MLS(Contender, Team) = \{(10, 2), (13, 5)\}$ . Here, both mappings 10 and 2 are to the tennis schema and mappings 13 and 5 are to the football schema.

Then using *MLS* and the function *S* as defined for the query above, we define the *CompetitionQuery* below.

*CompetitionQuery* =

$$\begin{aligned}
& (\pi_{Location, Date, Title, ((\tau_{Competition}(\alpha(Competition))) \\
& \quad Competition\_type, \\
& \quad Contender\_mid} \\
& \quad \bowtie_{Competition\_id=Competition\_id \wedge} (\alpha(Competes)) \\
& \quad \quad (Competition\_mid, Competes\_mid) \in S(Competition, competes) \\
& \quad \bowtie_{Contender\_id=Contender\_id \wedge} (\alpha(Contender))) \\
& \quad \quad (Competes\_mid, Contender\_mid) \in S(Competes, Contender) \\
& \quad \bowtie_{Title=Name \wedge} TeamQuery \\
& \quad \quad (Contender\_mid, Team\_mid) \in MLS(Contender, Team)
\end{aligned}$$

Here, we first query elements from the *Competition* domain structure in a similar fashion as above in query *TeamQuery*. We apply the *Competition*, *Competes*, and *Contender* domain structure types to their local schemas. We then join these relations on the *Competes* relationship and the *S* function defined above. We then project the attributes we need to create the widget as well as the *Contender\_mid* which we will use to join with *TeamQuery* (the query result for the team widget). We join *TeamQuery* with the results where the *Title* of the *Contender* is the same as the *Team Name* and where both the *Team* and the *Contender* have been mapped in the same local schema—so, we don't mix data from different local schemas. The query result can then be programatically nested to be shown by the competition widget in Figure 4 in the same fashion as it was for the team widget described above.

## 5. Related Work

In this paper we use a global-as-view model similar to traditional integration [3] but where traditional integration enforces a rigid singular global schema, we use many small global schema fragments (domain structures). Our domain structures can also be seen as abstract superclasses of the various local schema types to which the domain structures have been mapped similar to view integration and cooperation [2]. We extend these by bringing the local semantics through to the integrated functionality using our  $\tau$  operator.

We take inspiration from systems such as CLIO [4] for our mapping system. Like Clio, we want users to be able to create mappings as simply as drawing lines from local schemas to domain structures. Where we differ, is that we expect users to map local schemas many times to our small domain structures instead of trying to create entire schema mappings. This creates flexibility in how the domain structures may be later composed and means that end users need not understand every domain structure that could be mapped (only the domain structures of interest to the user). The flexibility of our mappings is also inspired by pay-as-you-go data integration such as that proposed by Madhavan [5].

Bringing local schema metadata to a global integration has been studied and developed in systems like SchemaSQL [6] and the Federated Interoperable Relational Algebra (FIRA) [7] and has been added to systems like Clio [8]. These systems address the problem that when integrating heterogeneous schemata it is often the case that data in one schema may exist as metadata in another schema (e.g., one schema may have city as an attribute of a company table whereas another schema may have one table for every city

the company has an office in). This is often exemplified by the use of the pivot/unpivot operation [9,10] to transform schema into data (unpivot) or data into schema (pivot). In contrast, we bring local schema metadata to our domain structures in order to bring the local semantics to the global level. We also attempt to lower the complexity of this by letting users add the local type operator to any domain structure type at any point in a query as simply as using another relational algebra operator. We believe this to be more intuitive than using database variables (in the case of SchemaSQL), having to deal with (possibly large) extraneous data as a result of the *down* operator in FIRA, or being limited solely to the attribute metadata in the case of pivot/unpivot.

As the usage of the semantic web [11] has grown, the number and variety of schemata within it has also increased, requiring the introduction of integration concepts long known in databases. Ontologies have replaced global schemas [12] and traditional integration techniques have been used, but again, this lacks the flexibility we require. In contrast, there has also been recent work into the use of small schemas (e.g. shallow or lightweight ontologies [1]) for use in search engines and other web integrations expressed in microformats<sup>2</sup>. The use of microformats requires that the schema elements are directly tied to the local data making it difficult to compose different schemas and requires editing the existing data to add global schema elements. These small schemas as well as larger ontologies have been used to create web widgets [13,14] similar to our widgets, but they are limited to presenting the data in the form of the schema.org or ontology schema whereas our widgets can bring local semantics through.

Our work is also inspired by the work on data modeling patterns such as those presented by Blaha [15], the Co-design and metastructure approach [16,17], and by ontology design patterns [18,19] in the semantic web. In practice, we see our domain structures as similar to patterns. But instead of using these patterns to develop schemas and systems as in the prior work, we instead use them to integrate heterogeneous schemata flexibly.

## 6. Conclusions and Future Work

In this paper we have described our preliminary work on information integration with local dominance. We have shown how domain structure types are defined and how they can be composed to create domain structures, which may be further composed with other domain structures. We have also shown how we can query the domain structures in order to integrate local databases by defining the apply operator and how we can bring local semantics to the domain level through the local type operator.

We plan to extend our formalism to incorporate more complex local schemas such as those that can be described using HERM [2] as well as incorporating these concepts into our domain structure types. This will allow us to better model web databases that often have complex attributes (nested attributes, sets, and lists) as well as handling higher order relationships—e.g. in the *Tennis* example above, it is more likely that the *Competes* relationship would have *Plays* and *TennisMatch* as its components since only students on the tennis team are likely to play in a tennis match. We plan to extend the HERM algebra as we have done with the relational algebra here. Using the more complex model provided by HERM we will be able to incorporate the output of our local type operators

---

<sup>2</sup><http://microformats.org/>

into the domain query answers possibly as nested attributes tying the domain structure type and the local type more closely than how it is currently presented simply as another attribute in the query answer linked to the domain structure type by a naming convention.

We also see the need to extend our current mapping formalism to incorporate tree/join paths. For example, users may wish to map a domain relationship to a join path in a local schema for a particular widget. Using our sports example we may wish to create a widget that lists all players in a sports league and in our local schema the league is associated with the team someone plays for instead of the player themselves.

In this paper we have referred to the relational data model, treating both the global schema and the local schemas as relational for the query language. Indeed, this is not really the case in practice, different models are used, and so it could be useful to integrate components that support translations from one model to another, for example as proposed in the MIDST project [20,21].

We have implemented a locally-dominant information integration system in a website to support educational materials<sup>3</sup> developed using the Drupal<sup>4</sup> content management system. The site integrates a number of different educational schemas (different types of course and content structures) and provides generic browse, download, and clone widgets. Our current clone widget allows data to be created using local semantics at the domain level. We are working on the formalism of this and the rules that apply when using traditional data manipulation language operations at a domain structure level.

## 7. Acknowledgments

We would like to thank Bernhard Thalheim for his expert guidance and support in this work. This work was supported in part by National Science Foundation, grant numbers 0840668 and 1250340. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

- [1] N. Shadbolt, T. Berners-Lee, and W. Hall, "The Semantic Web Revisited," *IEEE Intelligent Systems*, vol. 21, pp. 96–101, May 2006.
- [2] B. Thalheim, *Entity-relationship modeling: foundations of database technology*. New York: Springer, 1st ed., 2000.
- [3] M. Lenzerini, "Data integration: a theoretical perspective," in *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (New York, NY, USA), pp. 233–246, ACM, 2002.
- [4] R. J. Miller, L. M. Haas, and M. A. Hernández, "Schema Mapping as Query Discovery," in *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, (San Francisco, CA, USA), pp. 77–88, Morgan Kaufmann Publishers Inc., 2000.
- [5] J. Madhavan, S. R. Jeffery, S. Cohen, X. L. Dong, D. Ko, C. Yu, and A. Halevy, "Web-scale Data Integration : You can only afford to Pay As You Go," *World Wide Web Internet And Web Information Systems*, vol. 7, pp. 342–350, 2007.

---

<sup>3</sup><http://stemrobotics.cs.pdx.edu>

<sup>4</sup><http://drupal.org>

- [6] L. V. S. Lakshmanan, F. Sadri, and S. N. Subramanian, "SchemaSQL: An extension to SQL for multi-database interoperability," *ACM Transactions on Database Systems*, vol. 26, pp. 476–519, Dec. 2001.
- [7] C. M. Wyss and E. L. Robertson, "Relational languages for metadata integration," *ACM Transactions on Database Systems*, vol. 30, pp. 624–660, June 2005.
- [8] M. A. Hernández, P. Papotti, and W.-C. Tan, "Data exchange with data-metadata translations," *Proceedings of the VLDB Endowment*, vol. 1, pp. 260–273, Aug. 2008.
- [9] C. M. Wyss and E. L. Robertson, "A formal characterization of PIVOT/UNPIVOT," in *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05*, (New York, New York, USA), p. 602, ACM Press, Oct. 2005.
- [10] J. F. Terwilliger, L. M. L. Delcambre, D. Maier, J. Steinhauer, and S. Britell, "Updatable and evolvable transforms for virtual databases," *Proceedings of the VLDB Endowment*, vol. 3, pp. 309–319, Sept. 2010.
- [11] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.
- [12] N. F. Noy, "Semantic integration: a survey of ontology-based approaches," *ACM Sigmod Record*, vol. 33, no. 4, pp. 65–70, 2004.
- [13] E. Mäkelä, K. Viljanen, O. Alm, J. Tuominen, O. Valkeapää, T. Kauppinen, J. Kurki, R. Sinkkilä, T. Kansala, R. Lindroos, and Others, "Enabling the Semantic Web with Ready-to-Use Web Widgets.," in *FIRST*, pp. 56–69, 2007.
- [14] B. Nowack, "Paggr: Linked Data widgets and dashboards," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, pp. 272–277, Dec. 2009.
- [15] M. Blaha, *Patterns of Data Modeling*. CRC Press, June 2010.
- [16] H. Ma, R. Noack, K.-D. Schewe, and B. Thalheim, "Using Meta-Structures in Database Design," *Informatika*, vol. 34, pp. 387–403, 2010.
- [17] B. Thalheim, K.-D. Schewe, and H. Ma, "Conceptual Application Domain Modelling," in *Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM 2009)* (S. Link and M. Kirchberg, eds.), vol. 96 of *CRPIT*, (Wellington, New Zealand), pp. 49–57, ACS, 2009.
- [18] A. Gangemi and V. Presutti, "Towards a pattern science for the Semantic Web," *Semantic Web*, vol. 1, no. 1, pp. 61–68, 2010.
- [19] V. Presutti and A. Gangemi, "Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies," in *Conceptual Modelling - ER 2008* (Q. Li, S. Spaccapietra, E. Yu, and A. Olivé, eds.), vol. 5231 of *Lecture Notes in Computer Science*, (Berlin, Heidelberg), pp. 128–141, Springer Berlin Heidelberg, Oct. 2008.
- [20] P. Atzeni, L. Bellomarini, F. Bugiotti, F. Celli, and G. Gianforme, "A runtime approach to model-generic translation of schema and data," *Information Systems*, vol. 37, no. 3, pp. 269 – 287, 2012.
- [21] P. Atzeni, P. Cappellari, R. Torlone, P. A. Bernstein, and G. Gianforme, "Model-independent schema translation," *The VLDB Journal*, vol. 17, pp. 1347–1370, Nov. 2008.



# Modeling of classification error rate based on neural networks learners

Boštjan Brumen<sup>1</sup>, Ivan Rozman<sup>1</sup>, Aleš Černežel<sup>1</sup>

<sup>1</sup> University of Maribor, Faculty of Electrical Engineering and Computer science, Smetanova 17, Si-2000 Maribor, Slovenia

e-mail: <sup>1</sup>(bostjan.brumen, i.rozman, ales.cernezel)@uni-mb.si

## **Abstract.**

**Background:** A general, restrictions-free theory on performance of arbitrary artificial learners has not been developed yet. Empirically, not much research has been performed on the question of an appropriate description of artificial learner's performance.

**Objective:** The objective of this paper is to find out which mathematical description fits best learning curves produced by a neural network classification algorithm.

**Methods:** A Weka-based multilayer perceptron (MLP) neural network classification algorithm was applied to a set of datasets (n=109) from publicly available repositories (UCI) in step wise k-fold cross-validation and an error rate was measured in each step. First, four different functions, i.e. power, linear, logarithmic, exponential, were fit to the measured error rates. Where the fit was statistically significant (n=69), we measured the average mean squared error rate for each function and its rank. The dependent samples T-test was performed to test whether the differences between mean squared error rates are significantly different from each other, and Wilcoxon's signed rank test was used to test whether the differences between ranks are significant.

**Results:** The error rates, induced by a neural network, were best modeled by an exponential function. In a total of 69 datasets, exponential function was a better descriptor of error rate function in 60 of 69 cases, power was best in 8, logarithmic in 1, and linear in none out of 69 cases. Average mean squared error across all datasets was 0,000365 for exponential function, and was significantly different at P=0,002 from power, at P=0,000 from linear and at P=0,001 from logarithmic function. The exponential function's rank is, using Wilcoxon's test, significantly different at any reasonable threshold (P=0,000) from the rank of any other model.

**Conclusion:** In the area of human cognitive performance the exponential function was found to be the best fit for a description of an individual learner. In the area of artificial learners, specifically the multilayer perceptron, our findings are consistent with the mentioned. Our work can be used to forecast and model the future performance of a MLP neural network when not all data have been used or there is a need to obtain more data for better accuracy.

## **1. Introduction**

A mathematical description of human cognitive performance is well researched: the power function is generally accepted as an appropriate description in psychophysics, in skill acquisition, and in retention. Power curves have been observed so frequently, and in such varied contexts, that the term "power law" is now commonplace [1, 2]. In many real life situations the power law best fits the observed data [3]. However, several authors recently argued against the power law [4], explaining it holds only on an aggregate level; on specific learner's level the exponential law is advantageous.

On the other hand, artificial learners have not received such a high volume of research in terms of a description of their behavior [5]. Ideally, a description of a learning problem would be a functional dependency between the data, the learning algorithm's internal specifics and its performance (e.g. error). This way we could analytically determine the output (error rate) based on the input (data, selected learner). As of currently such a model has not been devised. Standard numerical (and other statistical) methods become unstable when using large data sets [6]. Different theoretical approaches provide estimates for the size of the confidence interval on the training error under various settings of the problem of learning from examples. Vapnik-Chervonenkis theory [7] is the most comprehensive description of learning from examples. However, it has some limits (e.g. oracle is never wrong) that make it difficult for real-life implementations, as described in detail [8]. Some results for a specific learner and for specific type of data can be found in the literature,

e.g. [9], but no general analytical solution is available.

In the absence of an accepted theoretical model, we can empirically measure a learner's performance on as many tasks (data sets) as possible. However, not much research was conducted on the description of performance of neural networks on a large scale comparison using several different data sets. Complementarily, there was some work on classification trees which can be predicted by a power law [10]. Several authors have either confirmed this or have been building on their results [11, 12]. But, a more recent research conducted by Singh has produced some evidence against the power law [13].

The research question of this paper is thus the following: which law (power, exponential, linear, or logarithmic) is a better mathematical description of an artificial learner, specifically multi-layer perceptron, over a larger number of available dataset? Our null hypotheses are as follows:

- The mean difference between function's  $f_i()$  and function's  $f_j()$  average mean squared error equals 0.
- The median of differences between function's  $f_i()$  and function's  $f_j()$  average rank equals 0.

Alternative hypotheses are that the mean squared error / median of differences are different.

The main contribution of this paper is the answer to the question: "Which mathematical description fits best a multilayer artificial neural network classifier?"

## 2. Method

We have chosen a multilayer artificial neural network classifier, which is freely available from the Waikato Environment for Knowledge Analysis (WEKA) project toolkit [14, 15] version 3.6.8, with standard built-in settings and initial values.

The computer used was equipped with Windows-7 (x64) operating system, an Intel i5-650 processor and 8 GB of DDR3 RAM. For statistical analyses we used IBM SPSS version 21.

### 2.1 Data collection

We used publicly available datasets from University of California at Irvine (UCI) Machine Learning Repository [16]. We selected the datasets where the problem task is classification; the number of records in a dataset was larger than 200 and the number of instances exceeded the number of attributes (i.e. the task was classification, not feature selection).

The UCI repository contains datasets in ".data" and ".names" format while Weka's native format is ARFF. Therefore we used files available from various sources, such as TunedIT [17], Håkan Kjellerstrand' weka page [18, 19] and Kevin Chai's page [20]. We gathered 121 datasets.

We used only the original or larger datasets where several ones were available and ignored any separate training or test set, or any associated cost model.

### 2.2 Data pre-processing

We followed the following steps for obtaining the error rate curve (i.e. learning curve) [21]:

1. Data items in a data set are randomly shuffled
2. First,  $n_{i=1}=50$  items are chosen
3. Build decision trees using k-fold cross-validation on sample size of  $n_i$  [22, 23];  $k$  was set to 10 [12, 22-24];
4. Measure the error rate for each tree in 10-fold run and average the result over 10 runs
5. Store the pair ( $n_i$ =sample size,  $e_i$ =error)
6. The number of items in a data set is increased by 10;  $n_{i+1}:=n_i+10$
7. Repeat steps 3-6 until all data items in a dataset are used.

## 2.3 Fitting a curve model to the measured data

The next step in our research was to fit a model to the error rate curves. We used four different functions, as in Equations 1-4:

power (POW):	$f(x) = p_1 + p_2 x^{p_3}$	Eq. 1
linear (LIN):	$f(x) = p_1 + p_2 x$	Eq. 2
logarithm (LOG):	$f(x) = p_1 + p_2 \log x$	Eq. 3
exponential (EXP):	$f(x) = p_1 + p_2 e^{p_3 x}$	Eq. 4

The functions do not have the same number of parameters ( $p_i$ ). They all include the constant  $p_1$  and coefficient  $p_2$ , in addition to potent  $p_3$  for the power and the exponential function. Based on the specifics of the problem and the speed of convergence we limited the parameters to the following intervals:

- $p_1$  to interval  $[0, 1]$  (error rate cannot be less than 0 and more than 1)
- $p_2$  to interval  $[0, 100]$  for power function and to  $[-100, 0]$  for the others, and
- $p_3$  to interval  $[-100, 0]$  (error rate is decreasing hence  $p_3$  needs to be negative)

We used the open-source GNU Octave software [25] and the built-in Levenberg-Marquardt's algorithm [26, 27], also known as the damped least-squares (DLS) method, for fitting the function parameters to the data.

The inputs to the algorithm were vector  $x$  (sample sizes  $n$ ), vector  $y$  (error rates  $e$ ), initial values of parameters  $p_i$  ( $[0,01; 1; -0,1]$  for POW,  $[0,1; -0,001]$  for LIN,  $[0,1; -0,01]$  for LOG and  $[0,01; 0,1; -0,01]$  for EXP), function to be fit to vectors  $x, y$  (power, linear, logarithm, or exponential), partial derivatives of functions with respect to parameters  $p_i$ , and limits of parameters  $p_i$  (as described above).

The algorithm's output were vector of functional values of fitted function for input  $x$ , vector of parameters  $p_i$ , where minimum mean squared error was obtained, and a flag whether the convergence was reached or not.

## 3. Results

For each dataset we tested the claim that the samples can be modeled by the probability density functions POW, LIN, LOG and EXP, respectively. We used the Pearson's chi-squared test ( $\chi^2$ ), also known as the chi-squared goodness-of-fit test or chi-squared test for independence, where the null hypothesis was  $H_0: r_\mu = 0$  or there is no correlation between the population and the model [28], at  $\alpha=0,05$ .

Out of 109 datasets, 69 were such that all the models can be used to describe the data (not shown). The remaining datasets are such that multilayer perceptron neural network algorithm does not capture their internal relations and cannot be used for classification, so we eliminated those from our further study.

From the vector of fitted function's values ( $f$ ) and from the vector  $y$  we calculated the mean squared error (MSE) of  $j^{\text{th}}$  dataset (DS), using Equation 5:

$$MSE_{DS_j} = \frac{\sum_{i=1}^n (y_i - f_i)^2}{n} \quad \text{Eq. 5}$$

where  $n$  is the number of input points, i.e. the size of a vector, for each individual data set  $DS_j$ . MSE describes how well the observed points fit to the modeled function. The average MSEs for each dataset are listed in Table 1, together with the rank of function's model. The model with lowest average MSE gets assigned rank 1. It can be seen that EXP is the best fit for the data in 60 of 69 cases, POW is best in 8 out of 69 times, LOG in 1 out of 69 cases, and LIN in none out of 69 cases. Average MSEs across all datasets were 0,000365 for EXP, 0,000417 for POW, 0,000517 for LOG, and 0,000588 for LIN.

**Table 1: Datasets and the average MSE across function models, and the model's rank (bold values indicate rank #1)**

Dataset	Average MSE (power)	POW rank	Average MSE (linear)	LIN rank	Average MSE (logarithm)	LOG rank	Average MSE (exponent)	EXP rank
ada_agnostic	0,000259	2	0,000278	4	0,000274	3	<b>0,000249</b>	<b>1</b>
ada_prior	0,000311	2	0,000387	4	0,000364	3	<b>0,000289</b>	<b>1</b>
analcatadata_broadwaymult	0,000483	2	0,000492	3	0,000498	4	<b>0,000457</b>	<b>1</b>
analcatadata_dmft	0,000738	2	0,000815	4	0,000772	3	<b>0,000712</b>	<b>1</b>
analcatadata_reviewer	0,000537	2	0,001051	4	0,000990	3	<b>0,000520</b>	<b>1</b>
anneal	0,000179	2	0,000391	4	0,000220	3	<b>0,000147</b>	<b>1</b>
australian	0,000858	4	0,000786	2	0,000846	3	<b>0,000780</b>	<b>1</b>
autos	0,001616	2	0,001993	3	0,002065	4	<b>0,001400</b>	<b>1</b>
badges_plain	0,000108	4	0,000097	2	0,000106	3	<b>0,000094</b>	<b>1</b>
balance-scale	0,000261	2	0,000387	4	0,000296	3	<b>0,000255</b>	<b>1</b>
baseball-hitter	0,000883	2	0,001102	3	0,002040	4	<b>0,000411</b>	<b>1</b>
baseball-pitcher	0,001733	4	0,001624	3	0,001572	2	<b>0,001521</b>	<b>1</b>
BC	0,001322	4	0,001288	2	0,001291	3	<b>0,001195</b>	<b>1</b>
biomed	0,000518	4	0,000456	2	0,000480	3	<b>0,000449</b>	<b>1</b>
breast-cancer	0,000358	4	0,000346	2	0,000356	3	<b>0,000345</b>	<b>1</b>
cars_with_names	0,000302	4	0,000262	2	0,000278	3	<b>0,000238</b>	<b>1</b>
CH	0,000162	2	0,000474	4	0,000210	3	<b>0,000115</b>	<b>1</b>
cps_85_wages	0,001058	2	0,001251	4	0,001067	3	<b>0,001045</b>	<b>1</b>
credit	0,000692	3	0,000732	4	0,000686	2	<b>0,000661</b>	<b>1</b>
csb_ch12	0,000174	3	0,000177	4	0,000171	2	<b>0,000164</b>	<b>1</b>
db3-bf	0,000628	4	0,000592	2	0,000620	3	<b>0,000566</b>	<b>1</b>
diabetes	0,000419	2	0,000424	4	0,000421	3	<b>0,000407</b>	<b>1</b>
ecoli	0,000491	4	0,000481	2	0,000487	3	<b>0,000471</b>	<b>1</b>
eucalyptus	<b>0,000502</b>	<b>1</b>	0,001112	4	0,000767	3	0,000542	2
eye_movements	0,000326	4	0,000319	2	0,000322	3	<b>0,000300</b>	<b>1</b>
genresTrain	<b>0,000150</b>	<b>1</b>	0,001083	4	0,000295	3	0,000254	2
gina_agnostic	0,000428	2	0,000612	4	0,000473	3	<b>0,000350</b>	<b>1</b>
gina_prior2	<b>0,000273</b>	<b>1</b>	0,001149	4	0,000576	3	0,000344	2
glass	0,002191	4	0,002136	2	0,002182	3	<b>0,002099</b>	<b>1</b>
heart-h	0,000450	4	0,000448	2	0,000450	3	<b>0,000442</b>	<b>1</b>
HO	0,000349	2	0,000520	4	0,000360	3	<b>0,000341</b>	<b>1</b>
hypothyroid	0,000223	4	0,000192	2	0,000199	3	<b>0,000182</b>	<b>1</b>
ionosphere	0,000248	4	0,000234	2	0,000242	3	<b>0,000233</b>	<b>1</b>
irish	0,000347	4	0,000216	3	0,000191	2	<b>0,000159</b>	<b>1</b>
jEdit_4.2_4.3	0,001174	2	0,002231	4	0,002158	3	<b>0,000894</b>	<b>1</b>
kc2	0,000273	2	0,000274	3	0,000276	4	<b>0,000269</b>	<b>1</b>
kr-vs-kp	0,000247	3	0,000645	4	0,000243	2	<b>0,000127</b>	<b>1</b>
kropt	0,000846	2	0,001512	3	0,001867	4	<b>0,000253</b>	<b>1</b>
landsat	0,000166	3	0,000160	2	<b>0,000159</b>	<b>1</b>	0,000356	4
letter	<b>0,000101</b>	<b>1</b>	0,001217	4	0,000520	3	0,000284	2
mfeat-factors	0,000131	2	0,000392	4	0,000212	3	<b>0,000127</b>	<b>1</b>
mfeat-fourier	<b>0,000251</b>	<b>1</b>	0,000500	4	0,000274	3	0,000261	2
mfeat-karhunen	0,000301	2	0,000925	4	0,000507	3	<b>0,000189</b>	<b>1</b>
mfeat-pixel	<b>0,000081</b>	<b>1</b>	0,000571	4	0,000226	3	0,000107	2
mozilla4	0,000111	2	0,000156	4	0,000126	3	<b>0,000091</b>	<b>1</b>
MU	0,000018	2	0,000028	4	0,000019	3	<b>0,000016</b>	<b>1</b>
mushroom	0,000051	4	0,000043	3	0,000029	2	<b>0,000018</b>	<b>1</b>
nursery	0,000145	3	0,000253	4	0,000083	2	<b>0,000056</b>	<b>1</b>
optdigits	0,000106	3	0,000197	4	0,000099	2	<b>0,000077</b>	<b>1</b>
page-blocks	0,000042	2	0,000106	4	0,000059	3	<b>0,000039</b>	<b>1</b>
pc4	0,000188	2	0,000224	4	0,000193	3	<b>0,000171</b>	<b>1</b>
pendigits	<b>0,000050</b>	<b>1</b>	0,000387	4	0,000158	3	0,000108	2
primary-tumor	0,000200	2	0,000761	3	0,001069	4	<b>0,000187</b>	<b>1</b>
prnn_synth	0,000088	3	0,000128	4	0,000088	2	<b>0,000079</b>	<b>1</b>
scopes-bf	0,000179	4	0,000165	3	0,000142	2	<b>0,000117</b>	<b>1</b>
SE	0,000217	4	0,000213	2	0,000216	3	<b>0,000211</b>	<b>1</b>
segment	0,000096	2	0,000205	4	0,000126	3	<b>0,000094</b>	<b>1</b>
sick	0,000069	4	0,000066	2	0,000067	3	<b>0,000065</b>	<b>1</b>
soybean	0,000411	3	0,000478	4	0,000381	2	<b>0,000344</b>	<b>1</b>
spambase	0,000236	2	0,000526	4	0,000360	3	<b>0,000235</b>	<b>1</b>
sylva_agnostic	0,000026	3	0,000033	4	0,000020	2	<b>0,000013</b>	<b>1</b>
sylva_prior	0,000059	4	0,000036	3	0,000026	2	<b>0,000017</b>	<b>1</b>
titanic	0,000217	2	0,000233	4	0,000218	3	<b>0,000202</b>	<b>1</b>
train	0,000325	3	0,000334	4	0,000322	2	<b>0,000305</b>	<b>1</b>
usp05	0,000285	4	0,000285	3	0,000284	2	<b>0,000269</b>	<b>1</b>
vehicle	<b>0,000990</b>	<b>1</b>	0,001331	4	0,001008	2	0,001036	3

Dataset	Average MSE (power)	POW rank	Average MSE (linear)	LIN rank	Average MSE (logarithm)	LOG rank	Average MSE (exponent)	EXP rank
VO	0,000114	2	0,000142	4	0,000124	3	<b>0,000099</b>	<b>1</b>
vowel	0,001244	2	0,001591	3	0,001623	4	<b>0,000584</b>	<b>1</b>
waveform-5000	0,000180	2	0,000281	4	0,000212	3	<b>0,000166</b>	<b>1</b>
<b>AVERAGE MSE</b>	<b>0,000417</b>		<b>0,000588</b>		<b>0,000517</b>		<b>0,000365</b>	
<b>RANK SUM</b>		<b>182</b>		<b>231</b>		<b>196</b>		<b>81</b>

As can be observed, the EXP had rank-sum of 81 and an average MSE of 0,000365. Please note that the rank is an ordinal value and hence calculating its mean value is inappropriate [28, p. 472].

Finally, the main research question was tested: which model was best? To rephrase, was EXP with the rank-sum of 81 and average MSE of 0,000365 significantly better than second-best POW with rank-sum of 182 and average MSE of 0,000417?

To test the significance of difference in MSE we used paired samples t-test for all combinations of models. The null hypotheses, the mean of differences between  $f_i$  (MSE) and  $f_j$  (MSE) equals 0, were as follows:

- $H1_0: \mu_{\text{MSE/power}} = \mu_{\text{MSE/linear}}$ ;
- $H2_0: \mu_{\text{MSE/power}} = \mu_{\text{MSE/logarithmic}}$ ;
- $H3_0: \mu_{\text{MSE/power}} = \mu_{\text{MSE/exponential}}$ ;
- $H4_0: \mu_{\text{MSE/linear}} = \mu_{\text{MSE/logarithm}}$ ;
- $H5_0: \mu_{\text{MSE/linear}} = \mu_{\text{MSE/exponential}}$ ; and
- $H6_0: \mu_{\text{MSE/logarithmic}} = \mu_{\text{MSE/exponential}}$ .

Because we conducted 6 comparisons, we used the Bonferroni correction to counteract the problem of multiple comparisons [29]. The correction is based on the idea that if an experimenter is testing  $n$  dependent or independent hypotheses on a set of data, then one way of maintaining the family-wise error rate is to test each individual hypothesis at a statistical significance level of  $1/n$  times what it would be if only one hypothesis were tested. We would normally reject the null hypothesis if  $P < 0.05$ . However, Bonferroni correction requires a modified rejection threshold for  $P$ ,  $\alpha = (0,05/6) = 0,008$ . Table 2 lists the results of statistical analysis for all six comparisons, with values in bold indicating significance at modified  $\alpha$  level.

Table 2: Paired samples t-test for MSE

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	POW (average MSE): LIN (average MSE)	-0,0001701	0,0002796	0,0000337	-0,0002373	-0,0001030	-5,055	68	<b>0,000</b>
Pair 2	POW (average MSE): LOG (average MSE)	-0,0000994	0,0002574	0,0000310	-0,0001613	-0,0000376	-3,209	68	<b>0,002</b>
Pair 3	POW (average MSE): EXP (average MSE)	0,0000521	0,0001345	0,0000162	0,0000198	0,0000844	3,219	68	<b>0,002</b>
Pair 4	LIN (average MSE): LOG (average MSE)	0,0000707	0,0002208	0,0000266	0,0000176	0,0001237	2,659	68	0,010
Pair 5	LIN (average MSE): EXP (average MSE)	0,0002222	0,0003187	0,0000384	0,0001457	0,0002988	5,792	68	<b>0,000</b>
Pair 6	LOG (average MSE): EXP (average MSE)	0,0001515	0,0003479	0,0000419	0,0000680	0,0002351	3,618	68	<b>0,001</b>

The results show that exponential function's average mean squared error is significantly different at any reasonable threshold from average MSE power ( $P=0,002$ ) linear ( $P=0,000$ ) and logarithmic function ( $P=0,001$ ), regardless if using the Bonferroni correction or not. Except for  $H4_0$ , all other hypotheses need to be rejected.

Additionally, we tested whether the ranks of functions are statistically significantly different from each other. We used related samples Wilcoxon's signed rank test. The null hypotheses, the median of differences

between  $f_i(rank)$  and  $f_j(rank)$  equals 0, were as follows:

- H7<sub>0</sub>:  $\mu_{1/2RANK / power} = \mu_{1/2RANK / linear}$
- H8<sub>0</sub>:  $\mu_{1/2RANK / power} = \mu_{1/2RANK / logarithmic}$
- H9<sub>0</sub>:  $\mu_{1/2RANK / power} = \mu_{1/2RANK / exponential}$
- H10<sub>0</sub>:  $\mu_{1/2RANK / linear} = \mu_{1/2RANK / logarithm}$
- H11<sub>0</sub>:  $\mu_{1/2RANK / linear} = \mu_{1/2RANK / exponential}$  and
- H12<sub>0</sub>:  $\mu_{1/2RANK / logarithmic} = \mu_{1/2RANK / exponential}$ .

Table 3 lists the results of Wilcoxon's signed rank test analysis for all six comparisons, with values in bold indicating significance at modified  $\alpha=0,008$  level.

Table 3: Wilcoxon signed rank test for different function models

Pair #	Pair	Sig. (2-tailed)
Pair 1	POW (rank) – LIN (rank)	<b>0,003</b>
Pair 2	POW (rank) – LOG (rank)	0,196
Pair 3	POW (rank) – EXP (rank)	<b>0,000</b>
Pair 4	LIN (rank) – LOG (rank)	<b>0,000</b>
Pair 5	LIN (rank) – EXP (rank)	<b>0,000</b>
Pair 6	LOG (rank) – EXP (rank)	<b>0,000</b>

The results show that exponential function's average rank is significantly different at any reasonable threshold from average rank of any other model ( $P=0,000$ ). Thus, all the above mentioned hypotheses H7<sub>0</sub> to H12<sub>0</sub> need to be rejected with exception of H8<sub>0</sub>.

## 4. Conclusion

In this paper we conducted an analysis of error rate curve produced by a selected multilayer perceptron neural network classifier. The results show that, in average, the best mathematical description of an artificial neural network learner is the exponential function. The results were consistent when using the mean squared error measure ( $P=0,000$  to  $0,002$  for t-test) and the rank assignment ( $P=0,000$  for Wilcoxon's test). Logarithmic, power and linear functions can, however, be superior in limited specific cases.

Since we observed a performance of an individual learner at 69 different tasks (data sets) we can conclude that our findings are consistent with the tests performed in the area of human cognitive performance, e.g. with works by Heathcote et al. [4].

The contribution of the work is important in many perspectives: firstly, the exponential model can be used to forecast the future performance of a neural network learner based on a small training sample. Secondly, early in the learning phase one can fit the model's parameters and estimate the final error rate. In case the estimated final performance is lower than the one required, one can modify the learner's parameters early in the process. Thirdly, the results of our experiment show that some datasets exist where modelling of the artificial learner's performance is not successful due to the inability of a learner to properly capture the data interrelations. This too could be detected early in the learning process.

## 5. Acknowledgements

This work was partially supported by the Slovenian Research Agency under grant number 1000-11-310138.

## 6. References

1. Anderson JR and Schooler LJ. Reflections of the Environment in Memory. *Psychological Science* 1991; 2(6): 396-408.
2. Anderson RB. The power law as an emergent property. *Memory & Cognition* 2001; 29(7): 1061-1068.
3. Clauset A, Shalizi CR and Newman MEJ. Power-Law Distributions in Empirical Data. *SIAM Review* 2009; 51(4): 661-703. DOI: doi:10.1137/070710111.
4. Heathcote A, Brown S and Mewhort DJK. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review* 2000; 7(2): 185-207. DOI: 10.3758/bf03212979.
5. Kotsiantis SB. Supervised Machine Learning: A Review of Classification Techniques. *INFORMATICA (Ljubljana)* 2007; 31(3): 249-268.
6. Dzemyda G and Sakalauskas L. Large-Scale Data Analysis Using Heuristic Methods. *Informatica (Lithuan.)* 2011; 22(1): 1-10.
7. Vapnik VN. Estimation of Dependences Based on Empirical Data. NY: Springer-Verlag 1982.
8. Brumen B, Jurič MB, Welzer T, Rozman I, Jaakkola H and Papadopoulos A. Assessment of classification models with small amounts of data. *Informatica (Lithuan.)* 2007; 18(3): 343-362.
9. Dučinskas K and Stabingiene L. Expected Bayes Error Rate in Supervised Classification of Spatial Gaussian Data. *Informatica (Lithuan.)* 2011; 22(3): 371-381.
10. Frey LJ and Fisher DH. Modeling decision tree performance with the power law. Seventh International Workshop on Artificial Intelligence and Statistics; 1999. San Francisco: Morgan Kaufmann.
11. Last M. Predicting and Optimizing Classifier Utility with the Power Law. 7th IEEE International Conference on Data Mining. ICDM Workshops 2007.; 2007. Omaha, Nebraska, USA: IEEE; DOI: 10.1109/icdmw.2007.31.
12. Provost F, Jensen D and Oates T. Efficient progressive sampling. Fifth International Conference on Knowledge Discovery and Data Mining; 1999. San Diego: ACM.
13. Singh S. Modeling Performance of Different Classification Methods: Deviation from the Power Law. Project Report. Nashville, Tennessee, USA: Vanderbilt University, Department of Computer Science; 2005.
14. Witten IH and Frank E. *Data Mining: Practical machine learning tools and techniques*. 2nd ed. San Francisco, CA, USA: Morgan Kaufmann; 2005. ISBN: 0120884070
15. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P and Witten IH. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter* 2009; 11(1): 10-18.
16. Asuncion A and Newman D. UCI Machine Learning Repository. 2010; Available from: <http://archive.ics.uci.edu/ml/datasets.html>. (Archived by WebCite® at <http://www.webcitation.org/6C2hgsRrX>).
17. TunedIT. TunedIT research repository. 2012; Available from: <http://tunedit.org/search?q=arff&qt=Repository>. Accessed: 2012-12-12. (Archived by WebCite® at <http://www.webcitation.org/6CqplN6Xr>).
18. Kjellerstrand H. My Weka page. 2012; Available from: <http://www.hakank.org/weka/>. Accessed: 2012-12-12. (Archived by WebCite® at <http://www.webcitation.org/6CqQ5pQtZ>).
19. Kjellerstrand H. My Weka page/DASL. 2012; Available from: <http://www.hakank.org/weka/DASL/>. Accessed: 2012-12-12. (Archived by WebCite® at <http://www.webcitation.org/6CqQcWpmy>).
20. Chai K. Kevin Chai Datasets. 2012; Available from: <http://kevinchai.net/datasets>. Accessed: 2012-12-12. (Archived by WebCite® at <http://www.webcitation.org/6CqQWlQEp>).
21. Brumen B, Hölbl M, Harej Pulko K, Welzer T, Heričko M, Jurič MB and Jaakkola H. Learning Process Termination Criteria. *Informatica (Lithuan.)* 2012; 23(4): 521-536.
22. Cohen PR. *Empirical methods for artificial intelligence*. Cambridge, MA, USA: MIT press; 1995. ISBN:
23. Weiss SM and Kulikowski CA. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. San Mateo, CA, USA: Morgan Kaufmann; 1991. ISBN:
24. McLachlan GJ, Do K-A and Ambrose C. *Analyzing microarray gene expression data*. Hoboken, NJ, USA: Wiley; 2004. ISBN: 0471226165
25. Eaton JW. GNU Octave. 2012; Available from: <http://www.gnu.org/software/octave/>. Accessed: 2012-12-12. (Archived by WebCite® at <http://www.webcitation.org/6CqyEvDKU>).
26. Marquardt DW. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics* 1963; 11(2): 431-441. DOI: 10.2307/2098941.

27. Levenberg K. A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly of Applied Mathematics* 1944; 2: 164–168.
28. Argyrous G. *Statistics for research: With a guide to SPSS*, 3rd ed. Thousand Oaks, CA, USA: SAGE Publications Ltd.; 2011. ISBN: 1849205957
29. Abdi H. The Bonferonni and Šidák Corrections for Multiple Comparisons. In: Salkind NJ, Editor. *Encyclopedia of Measurement and Statistics*. Thousand Oaks, CA, USA: SAGE Publications, Inc.; 2007. ISBN: 9781412916110



# Formation of a Collaborative Society

Ladislav BURITA <sup>a,1</sup> and Vojtech ONDRYHAL <sup>b</sup>

<sup>a</sup>*Department of CIS, University of Defence, Brno, Czech Republic and Department of IEIS, Tomas Bata University in Zlín, Czech Republic; [ladislav.burita@unob.cz](mailto:ladislav.burita@unob.cz).*

<sup>b</sup>*Department of CIS, University of Defence, Brno, [vojtech.ondryhal@unob.cz](mailto:vojtech.ondryhal@unob.cz)*

**Abstract.** The MilUNI knowledge portal, based on the knowledge base developed in ATOM software has been created at the authors' workplace with the aim to form a collaborative society of military universities. The analysis of the collaborative society concept is presented. The description of the MilUNI project is included. Some areas for university cooperation are proposed, as well as the measures facilitating the formation and development of the collaborative society.

**Keywords.** Collaborative society, military university, MilUNI, knowledge management system, ATOM

## Introduction

The Knowledge Management System (KMS) for the military universities (MilUNI) cooperation was created in summer 2012 at the Communication and Information Systems Department, Faculty of Military Technology, University of Defence, Brno, Czech Republic.

The topic was chosen as a suitable task for the internship of the French students from the university of ENSTA (École Nationale Supérieure de Techniques AVANCEES) Bretagne, Brest Engineering Institute, held at our department. The KMS research team at our department has considerable previous experience in the development of knowledge systems using ATOM (Aion Topic Maps engine) software (SW). Therefore, it was not difficult to prepare necessary prerequisites for the students practice.

The ontology was designed, which was gradually adjusted to better meet the objectives of the MilUNI. Students were gradually inserting the contents to the Portal from public sources. The result of their work is described in the article [1]. Further development was focused on the data check and including additional data, but especially on the Portal to ensure a pleasant user access to the system that was prepared in the final part of the project. The whole MilUNI system was ready for use in summer 2013 [2].

Studying and creating the KMS based on ATOM SW is also part of the university education. The students are introduced to the theory of KMS and work with knowledge. They are trained in the use of ATOM SW; they design their own ontology and create the application.

The article includes the analysis and explanation of the term 'collaborative society' and the method of its creation (Chapter 1); it describes a platform for cooperation of

---

<sup>1</sup> Corresponding Author.

military universities, MilUNI, (Chapter 2), provides an overview of potential areas of cooperation, (Chapter 3) and comments on attempts to create a collaborative society (Chapter 4).

## 1. Collaborative Society

To better understand the term collaborative let's compare the collaborative society with traditional ones [4]. The traditional groups and organizations tend to be structured vertically. Decisions are made at the top and people derive their influence and authority from their positions within the hierarchy. This is especially true in professional organizations where leadership is centralized, the work is mission-driven, processes are guided by procedures and statutes, and internal communication is mostly confined to departments, workgroups, and committees.

Collaborative groups, by contrast, are structured horizontally. Leadership, to the extent that it exists at all, is broadly distributed. Collaborative efforts tend to be loosely structured, highly adaptive, and inherently creative. Collaborative endeavours take many forms. Some common varieties include [4]:

- Public-private partnerships (sometimes referred to as social partnerships) are ad hoc alliances between otherwise independent organizations that span both the public and the private sectors;
- Future commissions, also known as search conferences, in which citizens and community leaders analyze trends, develop alternative scenarios of the future, and establish recommendations and goals for the community;
- Interagency collaborations aimed at improving social services to children, families, and other members of a community;
- Online networks designed to link various civic, educational, business, and governmental institutions within a community or region;
- School of University community partnerships designed to foster greater collaboration between schools, universities, and key community institutions;
- Networks and coalitions are loosely structured alliances among organizations, and citizens that share a commitment to a particular issue or place;
- Regional collaborative where local governments work together to promote economic development and service delivery.

Some questions to ask before embarking on a collaborative venture include:

- What are the structural relationships between the parties and the possible power issues inherent in the collaborative arrangement?
- Is there a clear understanding among all the parties of the respective goals and what form of leadership is required to facilitate the process?
- Does the project have some form of integrating structure, such as a cross-section of steering committees, to facilitate and coordinate decision-making and implementation?
- Will the project be more effective with a neutral, third-party mediator? Should the media be involved?
- Does the project have enough time, money, and staff support?

Building collaborative communities means finding new and better ways to work together. We need to create spaces where people can find each other, share ideas, and

discover common ground. The MilUNI project is an attempt to build such a collaborative society of military universities. Building a collaborative society is a time consuming and hard dynamic process. The method of building the society [4] consists of the three phases:

1. Problem setting phase.
2. Interest identification and setting the common goal.
3. Implementation phase.

### *1.1. Problem setting phase*

The parties must arrive at a shared definition of the problem, including how it relates to the interdependence of the various stakeholders and must make a commitment to collaborate. Other stakeholders, whose involvement may be necessary for the success of the endeavour, need to be identified.

The parties have to acknowledge and accept the legitimacy of the other participants, they must decide on what type of convener or leader can bring the parties together, and must determine what resources are needed for the collaboration to proceed.

### *1.2. Interest identification and setting the common goal*

This phase includes the following activities:

- Establishing ground rules and setting the agenda;
- Organizing subgroups, if the number of issues to be discussed is large or the number of people exceeds a dozen;
- Undertaking a joint information search to establish and consider the essential facts of the issue involved;
- Exploring of various alternatives and reaching agreement and settling for a course of action.

### *1.3. Implementation phase*

The implementation phase consists of the following tasks:

- Participating groups deal with their constituencies and parties garner support of those who will be charged with implementing the agreement;
- Structures for implementation are established; and finally the agreement is monitored and compliance is ensured.

## **2. Platform for MilUNI Collaboration**

The objective of the MilUNI is to provide a well-arranged platform for collaboration among military universities in teaching, research and exchanges of teachers and students. The system contains information about universities, their structure and focus of study. The university staff members are connected with recorded functions and activities, such as authorship of publications in conference proceedings and journals, and their participation in projects. There are full-text conference papers in the system, which enables the partners to study or quote them. The MilUNI is publicly available at <http://miluni.eu>.

The structure of the system is given by the ontology that consists of these main classes: university, organization, person, conference, collection and article. The MilUNI also includes information from the CIA World Factbook [6], a free source of information on countries of the world which is linked to other stored information. In this case it is the information about continents, countries, cities, and organizations. The system was developed within the research program of the Ministry of Defence [5].

### 2.1. Knowledge Base

The main feature of the MilUNI is a user friendly access to information about the structure of the universities, its main educational areas, research and conference activities, etc. The MilUNI data were collected from public sources on about 100 universities situated in 40 countries, 130 cities.

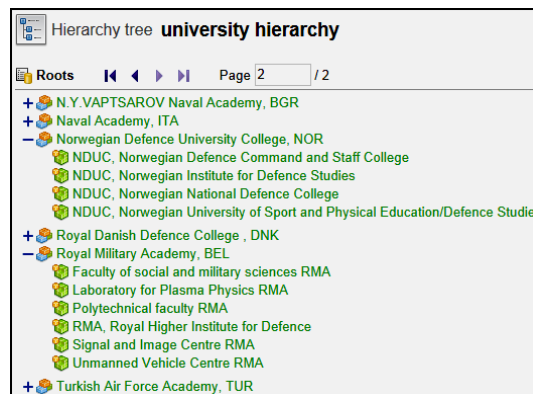


Figure 1. Universities hierarchy

The knowledge base (KB) is accessible and updatable through the ATOM Data editor, which is an environment for a skilled user in the knowledge system. The most common way to obtain the required information is by browsing the KB of a selected class, such as UNIVERSITY; see an example in Figure 2 that shows the university hierarchy.

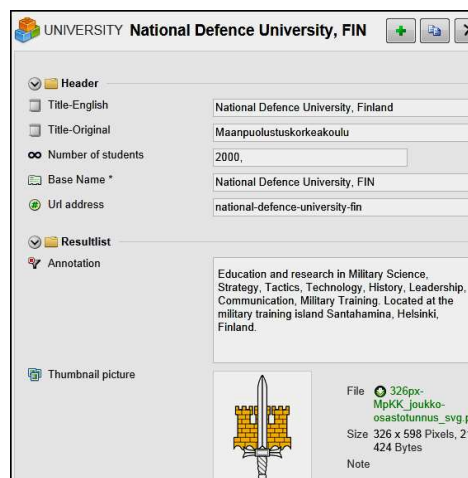


Figure 2. National Defence University, Finland

Required data about the university can be obtained (see Figure 2), then its field of study, the list of its academic staff and students, and perhaps even their publications at conferences.

## 2.2. Knowledge Portal

The knowledge portal (KP) covers the KB in the KMS to shield users from details of implementation. The KP is prepared as a typical portal template that is designed for any similar type of the KMS. The Portal structure and functions are designed with respect to the KB content and is connected with the KB (through ontology), so that it could be simply edited and personalized. The KP includes three types of pages (title, search result list and detail).



Figure 3. MilUNI – The title page

The title page of the KP consists of the registration box, main menu, news (search result area), full text search box, and results boxes, see Figure 3. The result list of the Finland military universities search list is shown in Figure 4, and the detail page about the National Defence University, Finland, is shown in Figure 5. Compare Figures 5 and 2 that visualize the detail about the same university, but in the different environments.



Figure 4. MilUNI – Search result: Military universities of Finland



Figure 5. MilUNI – Detail page: National Defence University, Finland

### 3. Areas of collaboration

Forming the community of cooperating universities is a long term process that should be supported at several university levels. It is true that there are some cases of successful cooperation, for example, between the University of Defense (UoD) in Brno, Czech Republic, and the Academy of the Armed Forces (AOS) in Liptovsky Mikulas, Slovakia, or between the UoD and Theresian Military Academy in Wiener Neustadt, Austria.

Also, some cases of a successful cooperation on the basis of the EU ERASMUS project are worth mentioning. At the Faculty of Military Technology (FMT), UoD, were the teachers from Bulgaria, Poland and Turkey, and, on the other hand, the teachers of the FMT were at the AOS or Military University of Technology in Warsaw, Poland. But still, these are only activities concerning individuals. It is the MilUNI project that should facilitate more intensive collaboration. The account of selected levels and areas of collaboration follows.

#### 3.1. Management of universities, faculties and departments

At the management level at universities, faculties and departments the administrative obstacles hindering cooperation need to be removed, and the conclusion of bilateral and multilateral agreements should be promoted. The cooperation also requires necessary funds. For example, the universities and their faculties concluded an ERASMUS Agreement, which states the structure and content of the mobility of students and teachers. However, this is only a necessary requirement for the implementation of the exchange of students and teachers.

#### 3.2. Joint projects and publishing support

The ideal forms of cooperation are joint projects. They require good personal cognition of partners who share a common, similar or complementary research interests. Now,

the MilUNI project can contribute to the development of such cognition. However, the cooperation from other parties on the MilUNI project is extremely desirable. It would be ideal if the data on their own university were operated by the same cooperative partner who knows the necessary data from their own university.

The general requirement for the university members is the preparation of their publications and the citation of them. There is another ambition of the MilUNI project lying in the recording of such information about conferences and selected articles. The articles could be available in full-text form, so that they could be studied or cited.

### *3.3. Exchange of teachers and students*

Exchange of teachers and students between universities should become an integral part of getting the appropriate teacher qualifications and student achievement of proper education. It is important for teachers not to be afraid of lecturing at other universities, and to spread the good name of their university and demonstrate their expertise on such occasions.

The ERASMUS program is a perfect opportunity for universities to organize teachers and students exchange. Although they might sometimes complain after returning back that not everything was absolutely perfect at the host university, or that sometimes it was not possible to fulfil everything that was planned, they confirm that they have never regretted this life and professional experience.

The MilUNI Portal supports the trips of teachers and students; it should help the actors to quickly contact you, to find a relevant university and to find out the necessary information about it.

## **4. Building the society**

The aim of creating the MilUNI Portal, as already mentioned, is to support the cooperation of military universities. However, it should be pointed out here that the MilUNI conception originated as an assignment for international students. The aim to promote military cooperation among universities emerged only later, as a quality and interesting result.

While the creators of the Portal had in mind in particular the cooperation of universities technical curricula, given that they belong to the Faculty of Military Technology, Department of CIS, in the process of MilUNI development the content was changed into including all military universities, which was related to the idea of creating a platform for cooperation.

We have to admit that our initial idea that just creating a portal accessible from the Internet will form the community automatically was naïve. It was a passive way to create a community. In the face of the demands on the community (see Chapter 1) this way of forming communities is inappropriate, as our experience confirmed.

When no one logged in after a certain time into the community, we understood the need to be more active to create the community. Each member of the team (6 persons) chose two or three universities. Using addresses on their Web sites, they contacted selected persons by an invitation letter with information about the objectives of the project and an appeal for cooperation. Also, the letter stipulated potential benefits for the university and its academic staff.

We waited for the response from the universities. From time to time, some contacts want us to explain something or ask us how to insert new data, but in general, there is no interest in working on the Portal even after 3 months. What is wrong? Are the persons responsible for cooperation between universities interested in new ways of communication? Is it an extra work load which the university staff member is not willing to accept? Did we address the right people? Despite all the proclamations, is the cooperation between universities less significant, and thus an unsupported matter? These were the questions that we were thinking about, but we did not find any relevant answers.

In addition to the above mentioned activities, the Portal has been introduced in professional journals, such as [1], [2] and at international conferences, such as [3]. The collaboration was also treated in the Erasmus Teaching Programme by prof. Burita at the Military University of Technology, Warsaw, Poland in May 2013. Concerning further work on the Portal, we are convinced that we will gradually and patiently upgrade and promote the Portal and its use at all levels of cooperation.

## Conclusion

The article deals with the problem of the creation of the military universities community, and it shows that it is not an easy task. It analyses the concept of the collaborative society. It describes a platform for cooperation between universities, the MilUNI Portal, and the objectives of its creation. Yet unsuccessful attempts to create a collaborative community within military universities are commented, and directions for the further research activities are proposed. There is an entirely appropriate concern that the creation of such communities is not a matter for the research team, but for the institution (organization) which is responsible for the cooperation. But we have not discovered persons responsible in any of the organizations, neither in the European Defence Agency (EDA), nor in NATO. The process we employed for the community creation does not follow exactly the process described in the first part of the article. It can also be the reason for the low success rate.

## References

- [1] BUŘITA, Ladislav; BROCHETON, Nicolas; BRUGET, Kévin; FERNANDES-LOPES, Mathieu. Knowledge Management System based on NATO Military Universities' cooperation for educational and research support. *Cybernetic Letters*, 2012, vol. 2012, no. 1, p. 1-5. ISSN 1802-3525
- [2] BUŘITA, Ladislav. Information Portal MilUNI for Military Universities Cooperation. *Cybernetic Letters*, 2013, vol. 1, no. 1, p. 1-4. ISSN 1802-3525.
- [3] BUŘITA, Ladislav. Knowledge Management System for Military Universities Cooperation. In: *International Conference on Military Technologies – ICMT 2013*. Brno: University of Defence, 2013, p. 501-506. ISBN 978-80-7231-917-6.
- [4] LONDON, Scott. Building Collaborative Communities. In *On Collaboration*. London: Tate, 2012. Available online <<http://www.scottlondon.com/articles/oncollaboration.html>>.
- [5] Research Program for Development of Organization CIS Department, Ministry of Defence, Prague, Czech Republic 2011-2015.
- [6] The World Factbook, Central Intelligence Agency, USA 2013, URL: <<https://www.cia.gov/library/publications/the-world-factbook>>.



# FOCAPLAS – A platform for cloud application development and running support

Xing CHEN\*, Keiichi SHIOHARA\*\*

*\*Department of Information & Computer Sciences  
Kanagawa Institute of Technology  
1030 Simo-Ogino, Atsugi-shi, Kanagawa 243-0292, Japan  
chen@ic.kanagawa-it.ac.jp*

*\*\*Department of Information & Computer Science  
Kanagawa Institute of Technology  
Atsugi-shi, Kanagawa, Japan  
s1385006@cce.kanagawa-it.ac.jp*

**Abstract.** Cloud computing is changing the utilization environments of computers in both enterprise and personal. Application development techniques and methodologies that are suitable to cloud environments are new challenging research topics. As demand for developing business applications is increasing rapidly and commercial profitability is dependent on decreasing the application development costs, it is essentially important to provide methods meeting the requirements of developing applications with low cost and short development time. Therefore, it is required to develop new methods facilitate the cloud application development based on easy-to-accomplish and end-user-composition. In this work, we compiled seven requirements of typical business application development such as data structure, database schema, page transition control, authorization, session management, programming, and input and output interface design. Furthermore we observe that none of current cloud development environments support a majority of these requested features. As a result, we present our own cloud application development platform, called FOCAPLAS that meets all of these requirements. A case study presenting a cloud application developing process is presented demonstrate how to use our FOCAPLAS. We believe that our requirements may serve as a valuable guide for cloud data modeling and our FOCAPLAS will be a useful platform for cloud application development.

## 1. Introduction

Cloud computing is changing the utilization environments of computers. Platforms such as Amazon Web Services, Google App Engine and Microsoft Azure are widely used for enterprise and personal purpose. Application providers use the resources of the platforms to run their applications for end users who use the applications. Cloud computing gives new environments to application providers with flexibilities on changing resource requirements for their applications and reducing application running cost.

Applications can be developed on three different cloud computing environments: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [1]. Developing applications on IaaS is similar to the traditional application developing method. PaaS provide developing resources that helps to reduce developing time and cost. Software functions of SaaS can be called through Web Application Interface (Web API). As the application running environments are quite different from the traditional running environments, it gives us challenging research topics to developing new cloud application development techniques and methodologies that are suitable to the cloud environments.

Cloud content services offer easy-to-accomplish contents creating methods for end users. These contents are commonly known as User Generated Contents (UGC). UGC are distributed in Blogs, Web pages and Wikis, etc. They supply contents' customers with environments consuming contents of the other users and creating contents for the other users. This accelerates the development of the cloud as it has huge amount of users generating many business chances. The business chances motivate the development of cloud applications in both enterprise and personal. As business chances come and go very quickly, it is essentially important to provide methods meeting the requirements of developing applications in low cost and short developing time. Therefore, it is required to develop new cloud application development methods facilitate the requirements of low cost and short development time based on easy-to-accomplish and end-user-compositions.

The most important application development environment to the application developers is that it helps to develop applications with low cost and short time. Cloud services are provided dedicating the application development. Developers use these services, typically through Web-based interaction for application development. It has recently proven to be attractive to many developers, as they decrease the application development cost and time. The application development services provide developing tools and data model for developers. Some services are designed to develop applications without needing the programming knowledge. Hence, end users who has no or less

programming knowledge can develop cloud applications. Yahoo! Pipes [2] and Google Enterprise Mashups [3] are examples of these kind of services.

In this paper, based on observations and considering the cloud application development requirements in Section 2, we first compile seven requirements of typical business application development that open the cloud data modeling to a wide range of developers in Section 3. Our cloud application development requirements address data structure, database schema, page transition management, authorization, session management, programming, and input output interface design.

Second, we briefly sketch FOCAPLAS in Section 4, our spreadsheet-based data modeling platform meeting all those requirements. Spreadsheet structure and functions supported by FOCAPLAS allow for data structure definition, logic control flow definition and input and output definition, etc. FOCAPLAS also serves cloud applications running same as those in today's cloud application servers. Finally, we will present a case study discussing a Web application development demonstrating how to use our method to develop cloud applications and give our conclusions.

## **2. Related work**

Both traditional programming development methods and non-programming methods can be utilized for developing cloud applications. Today's programming development Integrated Development Environments (IDEs) such as Microsoft Visual Studio [4] and Google App Engine [5] are widely utilized. Plenty of components, such as network related components and database components are provided to application developers. To decrease application developing times, templates for different applications are provided. The programming development methods have much developing flexibility, as developers can program desired behaviors of applications. In contrast, programming cloud applications requires a lot of knowledge and experiences on computer system, database and network, and involves lots of coding time and debug efforts. As high programming skill is required, it is difficult for developers who has less programming knowledge to develop cloud applications.

Google Enterprise Mashups [3] is an example of the non-programming methods which provides a graphic user interface (GUI) editor for creating Web applications. It is useful to integrate existed Web Applications and data. However, developers are heavily restricted to existed functions, e.g., it is difficult for developers to create their own applications as that of traditional programming methods. Furthermore, as functions of

user accounts management and authorization are not provided to the developers, it is greatly limited to business cloud application development. In order to implement user authorization and management, co-operation of programming specialists is required.

In the case of ExcelMashup [6], Microsoft published a mashup platform. Using this platform, developers can create and deploy applications embedding Microsoft Excel workbooks in Web applications and combining developers' code through JavaScript API. However, to use the platform, JavaScript programming skill and network knowledge are required. Furthermore, developers have to move away from open solutions to commercial Microsoft services, adapt to Microsoft-specific solutions and rewritten Excel applications.

Another Microsoft Excel based software package is provided by Microlab [7]. This software package runs on Microsoft windows server. Using this package, developers can move standalone Microsoft Excel based solutions to Web servers. It also provides functions connecting Excel cells to database records. However, its functions are not enough supporting to cloud application development.

These examples illustrate that although there are a lot of tools and platforms are provided for cloud application development, the supported range of functions are restricted and developers currently have to choose high cost programming methods. Furthermore, data modeling tools and platforms are not discussed and provided for cloud application developing.

### **3. Requirements**

Cloud computing is changing the utilization environments of computers in both enterprise and personal and providing many business chances. As these business chances come and go very quickly, it is essentially important to provide methods meeting the requirements of developing applications with low cost and short development time. In the following, we identify cloud application development requirements for easy-to-accomplish and end-user-compositions. We compiled seven requirements of typical business cloud application development such as data structure, database schema, page transition control, authorization, session management, programming, and input output interface design.

### **3.1 Data structure**

Using programming languages, developers are able to describe activities of applications following logic control flow and control data storage and network communication. As already mentioned, programming cloud applications requires a lot of knowledge and experiences on computer system, database and network, and involves lots of coding time and debug efforts. During the application development, developers have to mapping data models of application to database schemas. This includes tables and relations in databases with SQL programs which satisfy the requirements of the application if relational database model is used. After testing the databases, the deployment process consists of uploading data files, testing the application activities and communication interfaces between the applications and client machines. As plenty of components are provided by different providers, a developer who wants to use components provided by more than one providers has to map the data model to the data structures required by different providers. The developer also has to map the data model to Web page data structures for Web page load and transition. When user authorization is required, the user authorization database and program are also required to be created and coded. Additionally, developers are responsible for patching and updating the application.

For most developers, programming cloud application is too complex and raises high barriers. When developing cloud applications, it is ideal that developers can focus only on describing the data structures of applications. Thus, a cloud application development environment should free from programming and provide a data structure description tool where developers can easily describe their applications' data structures. This leads to a platform that appears to the developers supporting data structure description and having data processing functions. Current cloud development environments cannot satisfy this requirement. Thus, a new platform is required to be created for developing cloud applications.

### **3.2 Database schema**

Current cloud applications provide user data, time data, or geographic position data related services. These kind of services rely on database services. This leads to the definition of database schema. Plenty of research works are performed on this area and efficient methodology and tools are presented [8-16]. A famous model, the entity-relationship model (ER model) [17] is used for describing a database in an abstract way.

Schema integration is another important function required by cloud applications for combining data residing in different sources and providing users with a unified view of these data. It is basically required for the cloud application development platforms providing functions of describing databases and integrating different data sources.

To support connecting different data sources, a limited function, filter function, is provided. The filter function does not merges different data sources. It just passes data related to the selected key-field. Another example is a platform, Google Fusion Tables [18] that provides a merging function. Tables of Google Fusion Tables are like those of relational database model. Two tables can be merge based on an indicated key-field. However, Google Fusion Tables is not a cloud application development platform.

The cloud application development platform should support developers creating their own databases. Further, for a wider variety, the platform should support functions of merging and query.

### **3.3 Page transition control**

In general, different Web pages are used for receiving user inputted data and displaying result data. Web pages will be changed during an application running time. The term “page transition” represents changing the display of Web pages. The cloud application development platform should provide an easy-to-use to implement function for the page transition control. Unlike using programming development platform, it is not necessary to provide functions for controlling the page transition. In contrast, it should support developers writing different Web pages and defining relations between the pages. Detail program codes for the page transition should be generated by a program code generator of the platform. The cloud application development platform should support creating and managing Web page contents and controlling page transition. Developers write contents of Web pages and upload them to the platform. When new Web pages are uploaded, it is the platform's task to store and manage the Web pages and control the page transition. Developers should not have to know the details about the page transition control.

As contents of Web pages are changed by applications during running time, Web page transition is highly co-related to the contents. If main frame of a Web page is not changed, only parts of the contents are changed, such as calculation results, new Web page is not needed to be created. In such cases, to creating an ideal Web page, the platform should provide a mechanism for developers writing un-changed contents and changed

contents.

Providing the mechanism of merging un-changed and changed contents is also beneficial to the application developers. Thus, they can add calculation results of functions to a Web page without changing the other parts. The platform should also support developers writing different Web pages that will be indexed, stored and managed by the platform. Web page transition should be controlled by the platform.

### **3.4 Authorization**

Authorization is one of the main mechanisms required by business applications. Thus, the application development platforms should provide a function by using which developers can get authorized user information. The platform should also support developers uploading authorized user lists and querying user management databases. That is, the platform should provide user authorization query functions. If an application needs to access authorized data, the application development platform should automatically create a user registration database with fields of authorization page indexes and a query program. The platform should also give authorized result let developers to use it as input parameters of logic flow control functions.

If applications need uploaded authorized user lists, additional supports are required. The platform should support merging the uploaded user lists to the user management databases. Furthermore, when an uploaded list contains additional data fields that are not existed in the user management database, a new database should be created with a program describing the relation to the user management database.

### **3.5 Session management**

The previous sections addressed requirements of the application development on data structure, database and contents' displaying. This assumes that the platform provides a mechanism connecting a user's Web browser to relative databases and required contents. For applications developers, it is essentially required that each application's processing result will be sent back to the user who called the processing.

However, a cloud application will receive many calls from different browsers. Session identifier is commonly used to identify different browsers. A session identifier is a unique identifier which is sent from the server of the platform to a client browser. The

client browser stores and sends the identifier as an HTTP cookie in GET or POST queries. As the session identifier is the unique identifier for recognizing users who use the application, it is required to provide a mechanism to connect the user information to session identifiers. Furthermore, in order to recognize a user who opened browsers with different session identifiers, it is also required to provide a mechanism to connect cookie values with the user information.

Additionally, these mechanism should be provided to developers in a manner without the requirement for the developers to manage the session identification.

### **3.6 Programming**

The logic control flow is another requirement to the cloud application development platform. In order to develop applications without programming, commonly used functions should be pre-defined and generated. It is ideal that most applications can be developed by the pre-defined functions. For the applications that cannot be developed only by using pre-defined functions, logic control functions have to be used. Spreadsheet logic control functions, for example, the logic control functions of Microsoft Excel are wildly used by users even who have less programming knowledge. That is, if the cloud application development platform provides the logic control functions like those of spreadsheet, developers who have spreadsheet development knowledge can also develop cloud applications by using logic control functions. This leads to reduce application development costs because programming outsourcing is not required. Particularly, small companies can use the platform to develop business cloud applications flexibly because spreadsheet software packages are wild used in business processing. At the same time, by using spreadsheet functions to develop cloud applications, small companies can also modify and upgrade their cloud applications without the requirement of programming outsourcing.

There are several different non-traditional programming, or no-programming methods for application developing. Mashup platforms, such as Yahoo! Pipes [2] and Google Enterprise Mashups [3] represent a graphical model, where they offer a graphical editor to assemble blocks, called as gadgets or widgets. By assembling pre-created blocks by pipes same as the concept of UNIX shell pipeline, functions like aggregation, transformation, filter or sort can be performed. Other platforms like the ExcelMashup [6] and XCute [7] represent a spreadsheet based editor model with which applications can be developed in the way like standalone spreadsheet software packages. As a lot of business



data are spreadsheet based data, the spreadsheet based functions are naturally suitable for the business data processing.

### **3.7 Input and output interface design**

To achieve practical utilization of developed applications, the application development platforms should provide input and output interface design tool for developers. The tool should support two kinds of interfaces designing. The first one is the Web page design. The second one is the Web Application Interface (Web API) design. It is also required to the tools with efficient and flexible interface design functions which support developers to design different interfaces for organizations and companies.

As codes like JavaScript, Ajax, jQuery, CSS, etc. are utilized in Web page design, the application development platform should isolate those codes and provide a mechanism to use those codes as the third party designed interface. During the application runtime, the third party designed interface can be loaded and combined to the interface designed by the application developer. By providing this kind of mechanism, developers can develop applications without knowing details about JavaScript, Ajax, jQuery, etc.

## **4. FOCAPLAS**

FOCAPLAS is a cloud application development platform that we developed. We name it FOCAPLAS from the abbreviation of Formula Calculation Platform Service. It supports a spreadsheet based data modeling. By using FOCAPLAS, an application developer can build a cloud service by posting spreadsheet contents like posting document contents to a blog server. If developers want to update their applications, what is required is to post new spreadsheet contents.

We are building several cloud applications and testing the applications for validating the effectiveness of the platform. In our testing experiments, we built a computational application, a user authorization required application, a social networking application and a Web Application Program Interface (API) service [19]. Compared with the other applications and services similar to ours, the most important feature of ours is that our cloud applications and services are built based on spreadsheet contents without program scripts. The spreadsheet contents include formulas, Web page structure and page contents. Formulas are those of spreadsheet like formulas and those pre-defined formulas by

## FOCAPLAS.

FOCAPLAS is used in the way as follows. An application developer uploads spreadsheet contents to the platform. The uploaded formulas included in the spreadsheet contents are parsed on the server and executable program codes are generated automatically. When a user of the application connects to the server, a data submission form is sent to the user from the server side, or a client application-software connected to the server is called running by the server. Data submitted by users will be calculated based on the developer-uploading formulas. Calculating results will be sent back to the users directly or indirectly. Here, indirectly sending the calculating results means that user data are used to generate a query to search related data from database. In the services like the weight control service [19], recommendation services, etc., the retrieval results of databases, for example, comments or recommendations will be sent back to users.

In Figure 1, the structure of FOCAPLAS is illustrated. The platform is composed of a formula parser, a formula calculator, a database management system, data storages, an input/output device and interface specification interpretation equipment.

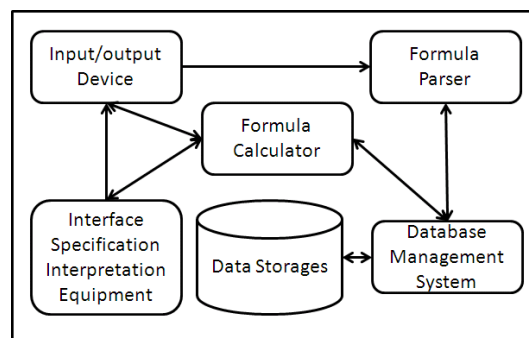


Figure 1. The structure of the formula Calculation Platform

The formula calculator receives the data sent by the service-users from the input and output device, computes received data and returns the calculation results back to the users. The formula parser parses formulas sent from service providers via the input/output device, generates executable codes and stores them into the database system.

The database management system is designed to provide functions of reading and storing six different kinds of data: (1) executable codes generated based on uploaded formulas, (2) user data, (3) user registration information, (4) response data, (5) user interface information and (6) Web API information. The interface device is designed for generating input and output forms or Web APIs defined by the serviced developers in the format of html, xml, etc. and sending them to the input/output device.

## **4.1 The Formula Parser**

The formula parser parses formulas posted by service providers and generates the following results: service Uniform Resource Locator (URL), input/output forms, application interfaces, and executable codes.

Developer-posting formulas are opened to publics as cloud services. The formula parser can parse the formulas uploaded in spreadsheet, csv and text formats. Developer-posting formulas are parsed by the formula parser and converted into program codes that can be executed by the formula calculator.

## **4.2 The Formula Calculation Service**

There are two types of formula calculation services. One is the kind of services that do not require user identification, such as “Converting Japanese calendar to the A.D. calendar”, “Loan calculation”, etc. The process of the service is performed in the way that user’s submitted data are calculated and calculation results are feedback to the users. During the calculation process, the user identification is not required.

Another one is the kind of services, which require user identification, such as “Cumulative intake of calories”, “Weight changes”. The processes of this kind of services rely on historical data submitted by users. During the service processing, when new user-submitting data is received, user’s previous submitted data is searched from the databases, and the current data is processed together with the previous data.

In section 5, we will show the latter case in details, in which user identification is required during the process of services. User identifications (IDs) and user IDs associated data can be obtained from FOCAPLAS. In the case study, we still present how to use these data. In order to provide the user identification related service, we use database formulas for writing user-submitting data into databases and reading them back from the databases based on user IDs.

## **4.3 The Database Management System**

The database manage system manages public databases and user personal databases. The public databases are accessed without the requirement of user identification. The user identification ID is required when a private database is accessed.

The database management system provides two kinds of management methods managing the private databases. The first kind of the method is to manage the database in which user IDs are used as the keywords during the database accessing. Another method is to manage individual user's databases. In this method, user's personal data are separately stored into different databases. When a user's personal database is accessed, an identification key is required.

In the case study presented in section 5, we select the latter method to develop a healthcare cloud service. A user's personal database is created when the user ID is created. If a user's personal database is not accessed during a period of times, it will be deleted. We will present the formulas for personal database creation and deletion in section 5.

## 5. A case study

In this case study, we develop a healthcare cloud service, referred to as a Weight Control Service, to show how to use FOCAPLAS. We utilize the mechanisms of the formula calculator and the database system of FOCAPLAS. We created the Weight Control Service to verify advantages of the platform. Based on our experiments, it is clear that cloud service with user identification can be developed without program coding. Developing time for the service is greatly reduced.

In order to protect user privacy, we selected the anonymous authentication. When a user connects our service at the first time, an anonymous user ID is created in the server side and sent to the user. The anonymous user ID is opened for mobile terminal application development. The system is also designed to accept manual inputted user ID. We also implemented data omission processing based on the user's previously submitted data. In the service, graphs of the weight changes are generated and sent to users.

### 5.1 Formulas for Creating the Service

We performed demonstration experiments to confirm the feasibility of FOCAPLAS. We prepared two types of interfaces, a Web API interface for mobile devices and a HTML interface for general-purpose Web browsers. In order to generate the anonymous user ID, we create a vector,  $\mathbf{V}$ , with 26 English uppercase characters.

$$\mathbf{V} = (A, B, \dots, Z) \quad (1)$$

Each element of the vector is defined by the formula  $\mathbf{V}(i)$ . In the formula, “ $i$ ” is an integer in the range from 1 to 26. For example,  $\mathbf{V}(1)$  indicates the letter “A” and  $\mathbf{V}(26)$  indicates the letter “Z”. An anonymous user ID is defined by a vector with “ $n$ ” elements. Each element is a randomly selected English uppercase letter. We use the following formula to generate anonymous user  $ID$ , where  $rand(1\dots 26)$  is a random number generator generating a number from 1 to 26.

$$ID = \sum_i^n v(rand(1\dots 26)) \quad (2)$$

We set “ $n$ ” to 6 in our service. User’s daily measured weight data are stored in the database system. In many cases, it is impossible letting a user input weight data every without omission. Various cases can be considered which will result to data omissions. For example, during business trips, it may be impossible to submit weight data. Or a user just does not want to submit the data because of overtime working. Considering these cases, data omissions are irregular. We use Newton polynomial [20] to interpolate the omitted weight data as it has a characteristic that the interpolated data can be reused for another interpolation. For the omitted dates, we use the linear interpolation [20] to interpolating the omitted dates with 1-day interval. Figure 2 shows an interpolation result. In the figure, weight data are submitted on July 1, 2 and 7. The omitted dates on July 4, 5 and 6 are interpolated.

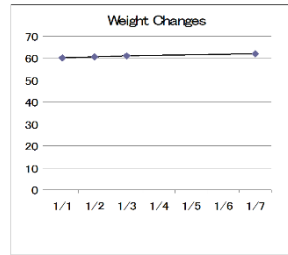


Figure 2. Interpolation result for 3 days

As shown in Figure 2, in order to reduce the amount of computation on interpolation calculation, the service is designed to store data up to 7 days.

If the user identification  $ID$ , which is separated from the user-submitting data, is not stored in the database, it will be stored into the database as a new  $ID$ . We define the user identification field as  $ID$ , and the user  $ID$  stored in the database as  $ID_d$ . We use the following formula to determine whether the user  $ID$  is stored in the database or not. If it is not stored in the database,  $ID_d$  is defined as  $ID_d = 0$ .

$$ID_{d \text{ Key}=ID_k} = Sel(ID) \quad (3)$$

If the user identification ID,  $ID_k$  is not stored in the database, that is,  $ID_d = 0$ , we use the following formula to store it into the database.

$$ID_{\text{Key}=ID_k \cap ID_d=0} = Sel^{-1}(ID_k) \quad (4)$$

In the formula, “ $\cap$ ” is defined as the logical product and  $ID$  is the field of user identification.

We use formula (5) to store user’s weight data into the database, where  $W_k$  is the user’s weight data and  $W$  is the field of weight data.

$$W_{\text{Key}=k} = sel^{-1}(W_k) \quad (5)$$

Formula (6) is used to read the weight data from the database.

$$W_k = Sel(W)_{\text{Key}=k} \quad (6)$$

## 5.2 Formulas for the Database Operations

Formulas for database creation and deletion are supported by FOCAPLAS. In order to create a database, a field name vector,  $\mathbf{N}$ , and a field character vector,  $\mathbf{C}$ , are required. The database creation formula is defined as “ $Cre$ ”.

$$Cre(\mathbf{N}, \mathbf{C}, name) \quad (7)$$

In the formula, “ $name$ ” is the name of the created database. When the vector  $\mathbf{N}$  or  $\mathbf{C}$  is an empty vector, the database “ $name$ ” will not be created.

The user personal database is crated in the case that the user identification ID,  $ID_k$  is not stored in the ID database ( $ID_d=0$ ). As we define all the user personal databases having the same fields, we define two common vectors,  $\mathbf{N}_{cm}$ ,  $\mathbf{C}_{cm}$ , for the database creation. After

the user identification ID,  $ID_k$ , is created, the user personal database will be created based on the following formulas, (8), (9) and (10).

$$\mathbf{N}_{ID_d=0} = \mathbf{N}_{cm} \quad (8)$$

$$\mathbf{C}_{ID_d=0} = \mathbf{C}_{cm} \quad (9)$$

$$Cre(\mathbf{N}, \mathbf{C}, ID_k) \quad (10)$$

Based on the formulas (8) and (9), for a user ID that is already generated and stored in the ID database, the vectors  $\mathbf{N}$  and  $\mathbf{C}$  are the empty vectors. Therefore, new user database will not be created.

The formula for database deletion is defined as “*Del*”. A personal databases will be deleted which are not accessed more than 7 days. In the “*Del*” formula, formula (11), a database name, name, is required.

$$Del(name) \quad (11)$$

In the formula (11), if name is an empty string, the function will not be executed.

We use the following formula to control the delete operation. In the formula, the user ID log is used. When an ID is not active more 7 days, the ID will be sent to the string variable,  $dn$ . When the no-empty string variable is sent to the formula *Del*, the database, which name is stored in the variable  $dn$ , will be deleted.

$$dn_{ID_k, \text{inactive log}=7} = ID_k \quad (12)$$

The personal database is deleted by using the following formula.

$$Del(dn) \quad (13)$$

### 5.3 Checking the Operation of the Created Cloud Service

The operations of reading and writing user identification ID, and user’s submitted weight data are check. The execution of the uploaded formulas for weight control is also checked.

In our experiments, we first checked whether a new ID key is created for the

beginning user or not, and whether the new ID key is stored in the database or not. After that, we confirmed whether the user-submitting data is stored under the ID key into the database or not. The operation check result is shown in Figure 3. We confirmed that a new ID key is created for the beginning user during the data submitting. For the subsequent data submitting, the user identification information is separated from the submitted data and stored into the database.

XOONUE		
recDate	myWeight	myMemo
2013/01/01	60	

Figure 3. The operation check result on new ID key generation

We also confirmed that new submitted weight data are correctly stored into the database under the already existed ID key. Figure 4 is the result of the operation check.

XOONUE		
recDate	myWeight	myMemo
2013/01/01	60	
2013/01/02	60.1	
2013/01/04	60.5	
2013/01/05	60.7	
2013/01/06	60.7	
2013/01/07	60.5	

Figure 4. New weight data are stored under the ID key

We confirmed interpolation for the data omissions. Figure. 5 is the result of the operation check. As shown in Figure 5, there is a data omission on January 3. The interpolation works correctly as shown in the figure. We also check the operation of the interpolation for the continuous data omissions. Figure 6 is the result of the operation check. There are data omissions on January 3, 4 and 5. As shown in the figure, the interpolation works correctly.

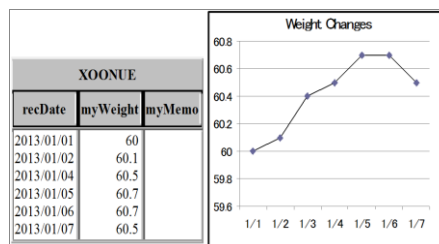


Figure 5. Interpolation for the one day omission



We verify the operation of the service used by mobile terminals. Figure 7 is the result of the operation check. The operation check was carried out by using Android mobile cell phones.

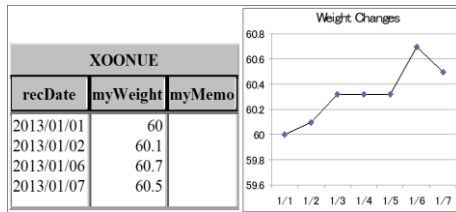


Figure 6. Interpolation for the continuous data omission

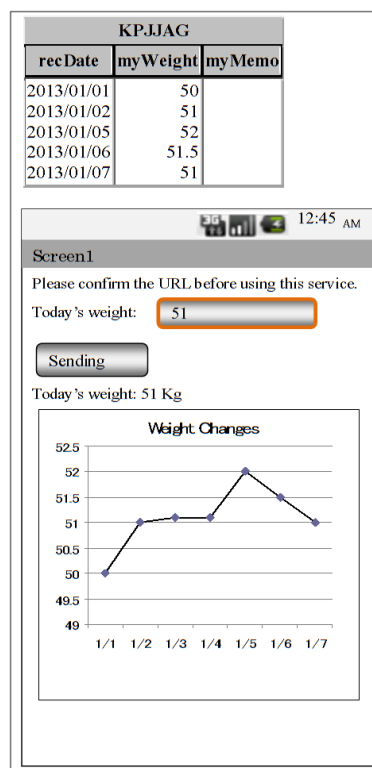


Figure 7. The operation check on an Android mobile cell phone

## 6. Conclusion and future work

Application development, especially the cloud application development demands are increasing rapidly because of shortened cycles of innovation and the spread of utilization of cloud computing and mobile devices. As commercial profitability is dependent on decreasing the application development costs, new cloud application

platforms are developed providing low-cost application development environments.

Despite the benefits of these cloud application development platforms, it can be observed that they are not commonly used for developing business cloud applications. This paper focuses on the business cloud application development. We identified that current cloud application development platforms cannot satisfy requirements of the business cloud application development. Moreover, application development tools supporting the cloud data modeling are needed to be developed. We presented seven requirements of typical business application development requirements such as data structure, database schema, page transition control, authorization, session management, programming, and input and output interface design. These requirements are presented for developers focusing their attentions to develop core parts of the application. Knowledge on technical details should not have to trouble developers. Instead, these should be supported and solved by cloud application development platforms. Further, the platform should automatically generate application program codes based on developers' data model description. It should provide a tool for data modeling which supports data model description, function description, input and output interface design, etc. As cloud applications are multi-user oriented, the platform should natively support user management and authorization. Keeping these requirements in mind we introduced FOCAPLUS, a cloud application development and running service platform. FOCAPLUS is a spreadsheet based data modeling platform. The platform supports cloud data modeling and automatically generates databases and program codes based on users' data model description. We also introduced a case study showing how to use our FOCAPLUS to developing cloud applications.

Our FOCAPLUS platform is currently at an early stage. Although it is now used for business cloud application development, there are still some limitations to our platform. These limitations restrict it opened to the public. We are currently working on solutions to eliminate them. Furthermore, we are working on providing new tools for cloud data modeling. This includes developing a new editor for cloud data modeling.

## 7. References

- [1] L. Youse, M. Butrico, and D. Da Silva, "Toward a Unified Ontology of Cloud Computing," In Grid Comp. Env. Workshop, 2008. GCE '08, Nov 2008.
- [2] Yahoo! Inc., "Yahoo! Pipes", <http://pipes.yahoo.com/pipes/>, (accessed 2013-12-20).
- [3] Google Inc., "Google Enterprise Mashups," <https://sites.google.com/site/>

fastonlinecontest/, (accessed 2013-12-20).

[4] Microsoft Corporation, "Microsoft Visual Studio," <http://msdn.microsoft.com/en-us/vstudio/aa718325.aspx>

[5] Google Inc., "Google App Engine," <https://developers.google.com/appengine/?hl=en>, (accessed 2013-12-20).

[6] Microsoft Corporation, "ExcelMashup," <http://www.excelmashup.com/>, (accessed 2013-12-20).

[7] Microlab Co. Ltd., "XCute", <http://www.microlab.jp/index.htm>, (in japanese), (accessed 2013-12-20).

[8] S. Ceri (ed.), "Methodology and tools for database design," North- Holland. Amsterdam, 1983.

[9] Chilson, D. and Kudlac, C., "Database design: A survey of logical and physical design techniques," Database Vol.15, No.1, 1983.

[10] Al-Fedaghi, S. and Scheuermann, P., "Mapping considerations in the design of schemas for the relational model," IEEE Trans. Softw. Eng. SE-7, 1 (Jan.), 1981.

[11] Batini, C. and Lenzerini, M., "A methodology for data schema integration in the entity relationship model," IEEE Trans. Softw. Eng. SE-10, 6 (Nov.), pp.650-663, 1984.

[12] Casanova, M. and Vidal, M., "Towards a sound view integration methodology," In Proceedings of the 2nd ACM SZGACTISZGMOD Conference on Principles of Database Systems (Atlanta, Ga., Mar. 21-23). ACM, New York, pp. 36-47, 1983.

[13] DAYAL, U., AND HWANG, H., "View definition and generalization for database integration in multibase: A system for heterogeneous distributed databases," IEEE Trans. Softw. Eng. SE-10, 6 (Nov.), 628-644, 1984.

[14] Demo, B., "Program analysis for conversion from a navigation to a specification database interface," In Proceedings of the 9th International Conference on Very Large Data Bases, VLDB Endowment, Saratoga, Calif, (Florence, Italy), pp. 387-398, 1983.

[15] Chiang, W., Basar, E., Lien, C. and Teichroew, D. "The view integration system (VIS)," ISDOS Rep. No. M0549-0, Ann Arbor, Mich., 1983.

[16] Pottinger P., & Berstein P., "Schema merging and mapping creation for relational sources," In Proceedings of the 11th international conference on extending database technology: Advances in database technology (EDBT '08), ACM, New York, pp.73-84, 2008.

[17] Chen, P.P. "The entity-relationship model: Toward a unified view of data," ACM

Trans. on Database Systems, Vol.1, No 1, pp. 1-36, March 1976.

[18] Google Inc., “Google Fusionables,” <http://www.google.com/drive/apps.html#fusionables>, (accessed 2013-12-20).

[19] Chen, X., Shiohara, K., & Tazumi, H. “Designing Formulas for Creating a Healthcare Cloud Service Based on a Formula Calculation Platform Service,” In ICIW 2013, The Eighth International Conference on Internet and Web Applications and Services, pp. 187-193, June 2013.

[20] Raymond W. Southworth, and Samuel L. Deleeuw, “Digital computation and Numerical Methods” McGraw-hill Book company, New York “Mathematics for computer II - Numerical analysis-”, KYORITSU SHUPPAN CO., LTD., pp.276-294, October, 1975, Japan (in Japanese)

# Comparison of Measurements of Learners' Performance

Aleš ČERNEZEL<sup>a</sup>, Boštjan BRUMEN<sup>a</sup>

<sup>a</sup> *University of Maribor, Faculty of Electrical Engineering, Computer Science and Informatics Smetanova 17, SI-2000 Maribor, Slovenia*

## Abstract.

*Background:* When assessing classifier's performance, multiple methods can be used. Regarding this, the  $k$ -fold cross-validation can be found as the most common method. Another such proposed method is the  $k$ -fold repeated cross-validation, which performs multiple repeats of the  $k$ -fold cross-validation and is claimed to improve (i.e. to lower) the variability of the measured performance.

*Objective:* In this paper, we compare the  $k$ -fold cross-validation method with the  $k$ -fold repeated cross-validation method. The objective of the paper is to experimentally prove that the  $k$ -fold cross-validation method is as good as the  $k$ -fold repeated cross-validation method.

*Methods:* Four classification algorithms (J48 decision trees, Multilayer Perceptron, Naïve Bayes and Support Vector Machines) were selected and applied on multiple datasets in the field of Life Sciences ( $n=35$ ) using both cross-validation methods. For statistical comparison, we used pairwise dependent Student's T-Test with the standard 95% confidence interval.

*Results:* The results of the statistical comparison between the cross-validation methods were as follows. Contrary to the findings of Kim [1], we found no significant difference between the two methods, regardless of the performed number of repeats (in the case of  $k$ -fold repeated cross-validation).

*Conclusion:* Since the  $k$ -fold repeated cross-validation method uses more computational effort and does not produce significantly different results, we can conclude that the  $k$ -fold cross-validation method is a better choice.

**Keywords.** Machine learning, classification, performance measurement, life sciences.

## 1. Introduction

In the field of supervised machine learning, multiple algorithms exist, each with its own fine-tuning settings. In order to measure the estimator performance, a method called cross-validation (CV) can be used [2]. With this method, a certain subsample, called the train sample, is used in order to train the classifier. After that, the classifier is tested on a (usually smaller) sample, called the test sample. Depending on the number of correct and incorrect classifications, certain metrics can be derived. The simplest one is classification error – the ratio between incorrect classifications and all classifications.

When performing CV, one can use multiple techniques. The most common one is the  $k$ -fold CV. Here, a sample is shuffled and divided into  $k$  parts. For each fold, i.e.

cross-validation, one of the parts is selected as the test set; others are used for the train set. The process is repeated  $k$ -times – each time a different part becomes the test set. When performing CV on a dataset subsample, the  $k$ -fold CV method subsamples it only once, potentially leaving out unseen instances. Therefore, one could argue that this can represent a certain bias.

Kim [1] proposed and empirically tested an improved  $k$ -fold CV method, called the  $k$ -fold repeated cross-validation (RCV). The method is simple: multiple subsequent runs of the standard  $k$ -fold CV method. In his paper, the selected number of repeats was five. Firstly, one could argue, that the selected number of repeats is too small. For instance, the proposed minimum number of repeats for the bootstrap method is 200, as suggested by Weiss and Kulikowski [3]. Secondly, he claims that the method improves (i.e. lowers) the variability of an estimator, but does not mention if there are any significant differences between the  $k$ -fold CV and the  $k$ -fold RCV regarding the accuracy in measuring estimator performance. Thirdly, his paper focuses on a single classification problem, which on top of all has only two possible decisions – a binary classification problem.

In this paper, we intend to upgrade the work of Kim [1] and assess the aforementioned weaknesses. We will perform an empirical experiment, using multiple datasets ( $n=35$ ) with different subsample sizes in the field of Life Sciences. We will compare the classification error between the  $k$ -fold CV and the  $k$ -fold RCV throughout all the datasets. In the case of RCV, we will also perform different number repeats (5, 50 and 100 repeats).

The main contribution of this paper is to find out which of the compared CV method is best to use regarding the accuracy of the classification error and the required computational effort.

The paper is organized as follows. We present the related work in Section 2. Here, we give an overview of the related work regarding comparison of different CV methods. In Section 3, we describe the scientific methods used in our experiment. In Section 4, we present and analyze the results. We conclude the paper with final remarks and comments in Section 5.

## 2. Related work

There are several papers comparing different CV methods. The most popular CV method is the  $k$ -fold CV. It is usually compared to the bootstrap method. The main differences between  $k$ -fold CV and the bootstrap method are: (1) bootstrap requires heavier computation than the  $k$ -fold CV (200+ repeats vs.  $k$  repeats) and (2) the bootstrap method performs instance selection in every repeat, while the  $k$ -fold CV performs it only once at the beginning.

Instance selection is crucial when learning on a subset of instances in a dataset. In this case, it is very important to select a representative sample. Failing to do so can result in obtaining inaccurate results. Given that the bootstrap method performs instance selection multiple times, the error when (possibly) selecting non-representative samples should even out. Thus, the bootstrap method gives less variable estimator, especially for smaller samples [4], [5].

Papers comparing the  $k$ -fold CV with the bootstrap method state that the comparison is not fair due to the large gap in the computational effort. Hence, researches try to even the computational effort by performing  $k$ -fold RCV.

In the works of Kim [1], the RCV method is said to be the general choice, although the bootstrap method can work better on some (smaller) samples. The most notable feature of the RCV method is, according to Kim [1], the reduced variability of the results. However, this claim is tested only on a single dataset and with five repeats in the case of the  $k$ -fold RCV method.

Borra & Di Ciaccio [6] performed a similar experiment, but instead used the repeated-corrected 10-fold RCV, proposed by Burman [7]. It was noted that it produced the most remarkable results and often outperformed other estimators.

### 3. Method

#### 3.1. The toolbox

In the experiment, four different classifiers were used. Namely J48 decision trees, Multilayer Perceptron (MLP), Naïve Bayes (NB) and Support Vector Machines (SVM). All selected classification algorithms were used with standard build-in settings and initial values. All the CV methods – both  $k$ -fold CV and  $k$ -fold RCV were performed using the Waikato Environment for Knowledge Analysis (WEKA) project toolkit version 3.7.6 [8].

Since some classification algorithms have different implementations, it is worth mentioning the exact implementations that were used in this experiment. For J48, the *J48* decision tree builder was used, which is based on Quinland's C4.5 tree induction [9]. For MLP, the *Multilayer Perceptron* [10] implementation was used. For NB, the *Naive Bayes* [11] implementation was used. For SVM, the John Platt's SMO (Sequential Minimal Optimization) implementation was used [12].

For the purpose of automating the machine learning process, a custom made Java application was built. The Weka Java API was used in order to access the classifiers and other utility features. The specifics of the said Java application will be discussed further in the paper.

The machine learning was performed on a virtual machine with 24 processor cores (Intel Xeon E5645 2.40 GHz) and 24 GB of memory. The operating system was Microsoft Windows Server 2008 R2. For statistical analyses, the IBM SPSS version 22 was used.

#### 3.2. Data collection

In the experiment, publicly available datasets from University of California at Irvine (UCI) Machine Learning Repository [13] were used. As aforementioned, only the datasets from the field of Life Sciences were chosen. The datasets were further filtered by problem task and number of instances. The selected problem task was classification and the dataset should have contained a minimum of 100 instances, but not more than 10,000 instances. The minimum limit was selected due to the further subsampling of the datasets and the maximum limit was selected due to limited computational resources.

Most of the datasets in the UCI repository are in “.data” and “.names” format, while Weka's native format is ARFF. In order to overcome this incompatibility we have used converted datasets from various 3<sup>rd</sup> party sources, such as TunedIT [14], Håkan Kjellerstrand' Weka page [15], [16] and Kevin Chai's page [17]. We used only

the original or larger datasets where several subsets were available and ignored any separate training or test set, or any associated cost model.

Despite multiple sources, not all selected datasets from the UCI repository were found in Weka's native format. These missing datasets were manually converted by the authors. The total number of gathered datasets is 35, listed in Table 1.

**Table 1:** Selected datasets with number of instances and attributes

<b>Dataset</b>	<b>Filename</b>	<b>Instances</b>	<b>Attributes</b>
Acute Inflammations	diagnosis.arff	120	6
Arrhythmia	arrhythmia.arff	452	279
Audiology (Standardized)	audiology.arff	226	69
Breast Cancer	breast-cancer.arff	286	9
Breast Cancer Wisconsin (Diagnostic)	wisconsin-diagnostic.arff	569	32
Breast Cancer Wisconsin (Original)	breast-w.arff	699	10
Breast Cancer Wisconsin (Prognostic)	wisconsin-prognostic.arff	198	34
Breast Tissue	breast-tissue.arff	106	10
Cardiotocography	cardiotocography.arff	2,126	23
Contraceptive Method Choice	cmc.arff	1,473	9
Dermatology	dermatology.arff	366	33
Echocardiogram	echocardiogram.arff	132	12
Ecoli	ecoli.arff	336	8
Fertility	fertility.arff	100	10
Haberman's Survival	haberman.arff	306	3
Heart Disease (Cleveland)	heart-c.arff	303	14
Heart Disease (Hungarian)	heart-h.arff	294	14
Hepatitis	hepatitis.arff	155	19
Horse Colic	colic.arff	368	27
ILPD (Indian Liver Patient Dataset)	ilpd.arff	583	10
Iris	iris.arff	150	4
Lymphography	lymph.arff	148	18
Mammographic Mass	mammographic_masses.arff	961	6
Mushroom	mushroom.arff	8,124	22
Parkinsons	parkinsons.arff	197	23
Pima Indians Diabetes	diabetes.arff	768	8
Primary Tumor	primary-tumor.arff	339	17
seeds	seeds.arff	210	7
Soybean (Large)	soybean.arff	307	35
SPECT Heart	spect.arff	267	22
SPECTF Heart	spectf.arff	267	44
Statlog (Heart)	heart-statlog.arff	270	13
Thyroid Disease	hypothyroid.arff	7,200	21
Yeast	yeast.arff	1,484	8
Zoo	zoo.arff	101	17

### 3.3. Machine learning

In the machine learning part, four classification algorithms were used on 35 datasets. Each dataset was further divided into four different sizes, called quartiles. The first quartile contains 25% instances from the dataset, the second 50%, the third 75% and the fourth contains all the instances. Having 35 datasets, divided into 4 quartiles, yields a total of 140 different configurations per classification algorithm.



**Figure 1:** The  $k$ -fold procedure [2]

- 
1. Shuffle instances in dataset.
  2. Divide the instances into  $k$  equally sized parts.
  3. Do  $k$  times:
    - a. Assign the  $k$ -th part as the test set.
    - b. Assign the other parts as the train set.
    - c. Train the classifier on the train set and test it on the test set. Record the classification error.
  4. Calculate the average of the  $k$  recorded classification errors.
- 

The procedure for the  $k$ -fold CV method is shown in Figure 1 [2]. Due to popular choice and explanation by Kohavi [18]  $k=10$  was chosen for the  $k$ -fold CV. For the number of repeats, multiple configurations were selected. In order to compare with the original paper by Kim [1], we have also decided to select  $n=5$  as the number of repeats. In our opinion, the  $n=5$  is too few for the number of repeats. Therefore to broaden the experiment and assess the weakness,  $n=50$  and  $n=100$  were also chosen. The total number of different CV method is therefore four: the plain  $k$ -fold CV and the  $k$ -fold RCV with three different configurations of repeats: 5, 50 and 100.

All CV methods have the same foundations and the only difference is in the number of repeats. Taking this into account, the basis for the custom Java application is simple. For each of the 140 configurations, we measured the performance of all the classifiers using the 10-fold CV multiple times – 101 times in our case. The reason for this number is as follows:

- For the plain 10-fold CV, the 101<sup>st</sup> repeat was used.
- For the 10-fold RCV with 5 repeats, the first 5 repeats were used.
- For the 10-fold RCV with 50 repeats, the first 50 repeats were used.
- For the 10-fold RCV with 100 repeats, the first 100 repeats were used.

One may notice that we were adding repeats to the 10-fold RCV cases, whereas the plain 10-fold CV had its own separate repeat. There are two reasons for this decision: (1) cases between 10-fold CV and 10-fold RCV must be independent and (2) in the cases of 10-fold RCV with different numbers of repeats, we wanted to observe if adding more repeats produces better estimates for the classification error.

The final result of machine learning is a classification error for each algorithm and for each of the 140 configurations. In the case of 10-fold CV, a single value was obtained. In the case of 10-fold RCV, multiple values were obtained and were averaged in order to be comparable with the plain 10-fold CV.

### 3.4. Comparison of cross-validation methods

In the machine learning process, four different CV methods were used: one plain 10-fold CV and 10-fold RCV with three different numbers of repeats (5, 50 and 100). Given 140 different configuration per classification algorithm, there are a total of 560 cases per CV method. The values are numerical and represent the classification error.

In order to statistically compare all four CV methods across all cases, the dependent Student's T-Test for paired sample was used. The main idea behind this test is to find out whether the differences between two related samples are significant or not. The null hypothesis for this statistical test is the following: there are no differences between the samples; or in other words: the difference equals zero ( $H_0: \mu_D = 0$ ).

Because we can only compare two samples at once, multiple pairwise comparisons are needed. In our case, the number of pairs equals 6. All pairwise tests were conducted

using the IBM SPSS version 22 and the standard 95% confidence interval was used. The results will be presented in the next section.

#### 4. Results

Conducting the dependent Student's T-Test for paired sample produces multiple tables, containing various test statistics. The statistics regarding basic information of each sample (N, mean, std. deviation, and std. error mean) and correlations between the pairs will be omitted, as they are not relevant in this experiment. However, the statistics regarding the differences between each pair are presented fully in Table 2.

**Table 2:** Paired samples test statistics

#	Pair	Mean	Std. dev.	t	df	Sig. (2-tailed)
1	CV-RCV5	0.000325	0.030348	.253	559	0.800
2	CV-RCV50	-0.000595	0.029521	-.477	559	0.633
3	CV-RCV100	-0.000887	0.029388	-.714	559	0.475
4	RCV5-RCV50	-0.000920	0.013269	-1.641	559	0.101
5	RCV5-RCV100	-0.001212	0.013799	-2.079	559	0.038
6	RCV50-RCV100	-0.000292	0.002834	-2.437	559	0.015

The pairs in the table are labeled as follows: the plain 10-fold CV is labeled as CV. The 10-fold RCV with 5, 50 and 100 repeats are labeled as RCV5, RCV50, and RCV100, accordingly. The column "Mean" represents the mean difference between the samples in the pair throughout all the cases. Similarly, the column "Std. dev." represents the standard deviation of the differences between the cases in the pairs. The column "t" represents the value of the Student's T-Test and the column "Sig. (2-tailed)" represents the p-value considering to the degrees of freedom (column "df").

The values of interest are the p-values, shown in the last column. If the value is below .05, then the null hypothesis can be rejected – meaning there are significant differences between the two samples. In other words, one of them is better than the other and the "Mean" column is the key to determine which one.

One can observe that we have failed to reject the null hypotheses in all but two cases – pair #5 and #6. But given the fact that multiple hypotheses are being tested, a possible Type I error must be taken into account. In order to address this matter, the Holm-Bonferroni method [19] was used, which resulted that we failed to reject even cases #5 and #6. With this in mind, we can conclude that there are no differences whether we use the plain  $k$ -fold CV or the  $k$ -fold RCV, regardless of the number of repeats that were used in our experiment.

But since the p-values in pair #5 and #6 are significant if they are taken out of the context (without other hypotheses and the use of the Holm-Bonferroni method), one should take a closer look at why this has happened and what does it mean. Firstly, we need to understand the way the dependent Student's T-Test works. The equation for calculating the t value is relatively simple and self-explanatory. It divides the "Mean" value with the "Std. dev." value, which is further divided by the square root of the number of the sample size, as shown in equation ( 1 ).

$$t = \frac{\bar{x}_D}{s_D/\sqrt{n}} \quad (1)$$

In the case of pair #6, the mean difference and standard deviation is one order of magnitude smaller than in other pairs. This is consistent with findings of Kim [1],

saying that the RCV method reduces the variability of the results. Especially when performing a larger number of repeats. In our case, this is manifested as a low standard deviation. Also, the mean difference between the two samples is .03%, the lowest of all pairs. As for pair #5, the standard deviation is similar as in pair #4, but there is a relatively huge difference between the mean values (approx. 30%). This is the main reason why pair #5 produces a significant p-value and pair #4 does not.

But since there are no differences between the  $k$ -fold CV and the  $k$ -fold RCV across all cases, there might be a difference when considering only the smallest subsamples (quartile = 1), where the variance of measurements is expected to be the highest. In this case, the RCV method shows its strengths as it performs subsampling multiple times and thus avoids possible bias and reduces variance as claimed by Kim [1]. Here again, it is to notice that Kim [1] used only one dataset with only five repeats in order to make such a claim.

In order to test this, we have performed the dependent Student's T-Test same as before, but applied a filter on the cases in such a way, that only the cases with *quartile* = 1 were selected. The number of cases dropped from 560 to 140. The results are presented in Table 3, which has the same structure as Table 2, so the explanation of the meaning of the columns will be omitted.

**Table 3:** Paired samples test statistics with case selection (*quartile* = 1)

#	Pair	Mean	Std. dev.	t	df	Sig. (2-tailed)
1	CV-RCV5	-0.004383	0.049096	-1.056	139	0.293
2	CV-RCV50	-0.006297	0.048453	-1.538	139	0.126
3	CV-RCV100	-0.007033	0.048353	-1.721	139	0.087
4	RCV5-RCV50	-0.001914	0.021520	-1.052	139	0.294
5	RCV5-RCV100	-0.002650	0.022357	-1.403	139	0.163
6	RCV50-RCV100	-0.000736	0.004286	-2.033	139	0.044

The results are still not significantly different. Here, only the pair #6 managed to have a p-value of less than .05. But again, when considering possible Type I errors and applying the Holm-Bonferroni method, all cases fail to reject the null hypotheses. If both tables are compared pairwise, one can observe that there is a difference of one order of magnitude for the "Mean" column in most cases. The difference in the column "Std. dev." is also noticeable, but within one order of magnitude. This means that the smaller subsamples produce more variability, which is consistent with findings of Kim [1].

To sum up, the  $k$ -fold RCV is superior to the  $k$ -fold CV method when considering the results variability. However, observing the difference between the two methods yields insignificant results according to the dependent Student's T-Test. In other words, the  $k$ -fold CV method is as good as the  $k$ -fold RCV method and in addition it requires less computational effort.

## 5. Conclusions

In this paper, we statistically compared two different cross-validation methods, namely  $k$ -fold cross-validation (CV) and the  $k$ -fold repeated cross-validation (RCV). We used four classification algorithms on multiple publicly available datasets ( $n=35$ ) from the field of Life Sciences. Each dataset was further divided into four differently sized subsamples.

The basis for the performed experiment is as follows. When performing CV on a dataset subsample the  $k$ -fold CV method subsamples it only once, potentially leaving out important instances that could otherwise be seen by the bootstrapping or when using the whole dataset. The main contribution of this paper is the comparison of the CV methods regarding the accuracy of the classification error and the required computational effort.

A similar experiment was conducted by Kim [1] where the  $k$ -fold RCV method was considered superior regarding the variance, but there was no comparison of the measured performance between various CV methods. It is also to notice that in the experiment by Kim [1] only one dataset was used and the number of repeats in the  $k$ -fold RCV method was five.

In our experiment, four CV methods were compared: 10-fold CV and 10-fold RCV with 3 different numbers of repeats (5, 50 and 100). The results from comparison of the two CV methods show that there is no significant difference between them, regardless of the number of the performed repeats. We further analyzed the data and compared only the cases with the smallest subsamples, where the  $k$ -fold RCV method should perform better. But again, there was still no significant difference between the two CV methods.

Our findings regarding the variability of the measured performance are consistent with Kim [1] – the  $k$ -fold RCV method showed less variability than the  $k$ -fold CV method. However, the difference in the measured performance is not significantly different. In other words, the additional computation effort required by the  $k$ -fold RCV does not make any difference and is therefore redundant.

## 6. Acknowledgements

This work was partially supported by the Slovenian Research Agency under grant number 1000-11-310138.

## 7. References

- [1] J.-H. Kim, ‘Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap’, *Computational Statistics & Data Analysis*, vol. 53, no. 11, pp. 3735–3745, Sep. 2009.
- [2] P. R. Cohen, *Empirical methods for artificial intelligence*. Cambridge, MA, USA: MIT Press, 1995.
- [3] S. M. Weiss and C. A. Kulikowski, *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. San Mateo, Calif.: M. Kaufmann Publishers, 1991.
- [4] B. Efron, ‘Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation’, *Journal of the American Statistical Association*, vol. 78, no. 382, pp. 316–331, Jun. 1983.
- [5] B. Efron and R. Tibshirani, ‘Improvements on Cross-Validation: The .632+ Bootstrap Method’, *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 548–560, Jun. 1997.

- [6] S. Borra and A. Di Ciaccio, 'Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods', *Computational Statistics & Data Analysis*, vol. 54, no. 12, pp. 2976–2989, Dec. 2010.
- [7] P. Burman, 'A Comparative Study of Ordinary Cross-Validation, v-Fold Cross-Validation and the Repeated Learning-Testing Methods', *Biometrika*, vol. 76, no. 3, pp. 503–514, Sep. 1989.
- [8] I. H. Witten, *Data mining: practical machine learning tools and techniques*, 2nd ed. Amsterdam ; Boston, MA: Morgan Kaufman, 2005.
- [9] J. R. Quinlan, *C4.5: programs for machine learning*. San Mateo, Calif: Morgan Kaufmann Publishers, 1993.
- [10] Malcolm Ware, 'WEKA Documentation'. University of Waikoto.
- [11] G. H. John and P. Langley, 'Estimating Continuous Distributions in Bayesian Classifiers', in *Eleventh Conference on Uncertainty in Artificial Intelligence*, San Mateo, 1995, pp. 338–345.
- [12] J. Platt, 'Fast Training of Support Vector Machines using Sequential Minimal Optimization', in *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola, Eds. MIT Press, 1998.
- [13] K. Bache and M. Lichman, 'UCI Machine Learning Repository'. University of California, Irvine, School of Information and Computer Sciences, 2013.
- [14] 'Repository of data sets and algorithms', *TunedIT*, 2013. [Online]. Available: <http://tunedit.org/repo> (Archived by WebCite® at <http://www.webcitation.org/6CqplN6Xr>).
- [15] H. Kjellerstrand, 'My Weka page', 2013. [Online]. Available: <http://www.hakank.org/weka/> (Archived by WebCite® at <http://www.webcitation.org/6Cqq5pQtZ>).
- [16] H. Kjellerstrand, 'My Weka page/DASL', 2013. [Online]. Available: <http://www.hakank.org/weka/DASL/> (Archived by WebCite® at <http://www.webcitation.org/6CqqCwPmy>).
- [17] K. Chai, 'Datasets', *Kevin Chai's Homepage*. [Online]. Available: <http://kevinchai.net/datasets> (Archived by WebCite® at <http://www.webcitation.org/6CqqWlQEp>).
- [18] R. Kohavi, 'A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection', in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, San Francisco, CA, USA, 1995, pp. 1137–1143.
- [19] S. Holm, 'A Simple Sequentially Rejective Multiple Test Procedure', *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, Jan. 1979.

# Event ontology specification based on the theory of valency frames

Martina Číhalová

*Palacky University of Olomouc,  
Department of Philosophy, Křížkovského 511/10, 771 47 Olomouc, Czech Republic*

**Abstract.** Linguistic theory of verb-valency frames is applied to the analysis of event and process ontology from the point of view of agents' reasoning. Since *logical* analysis presupposes full *linguistic* competency, it is suitable to make use of the results of linguistic analysis, in particular of verb-valency frames. Each process can be specified by a verb (*what* is to be done), possibly with parameters like the agent/actor of the process (*who*), the object to be operated on, resources, etc. In verb-valency frames each verb is characterized by the participants of an action denoted by the verb. Using verb-valency frames we can thus obtain a fine-grained specification of a process/procedure. The novel contribution of this paper is a proposal of a process ontology based on the results of linguistic analyses and classifications. Particular types of participants are then assigned to processes as their requisites or typical properties. The specification tool is Transparent Intensional Logic (TIL) with its procedural as opposed to denotational semantics.

**Keywords.** Verb-valency frame, event, action, Transparent Intensional Logic, verb participants

## 1. Introduction

The term 'ontology' has been borrowed from philosophy, where ontology is a systematic account of existence. In recent computer science and artificial intelligence a formal ontology is an explicit and systematic conceptualization of a domain of interest. Given a domain, ontological analysis should clarify the *structure* of knowledge on what exists in the domain. A formal ontology is, or should be, a stable heart of an information system that makes knowledge sharing, reuse and reasoning possible. As J. Sowa says in [1, p. 51], "logic itself has no vocabulary for describing the things that exist. Ontology fills that gap: it is the study of existence, of all the kinds of entities – abstract and concrete – that make up the world".

In this paper a method of building up ontology of processes is proposed. While ontologies of a given specific domain are frequently studied, ontologies of *processes* have been rather neglected. The goal of this paper is to fill the gap and propose a method for the specification of *process ontology*.

Nowadays, there are many process specification tools and languages. For instance, these are well-known standards: *WMC* (Workflow Management Coalition)<sup>1</sup>, *BPMN*

---

<sup>1</sup> For details see [2].

(Business Process Model and Notation)<sup>2</sup>, *VPML* (Visual Processing Modelling Language)<sup>3</sup>, *PSL* (Process Specification Language)<sup>4</sup> and *RUP* (Rational Unified Process)<sup>5</sup>. In [6] we provided a brief summary of these tools and a comparison of concepts these languages use for modelling processes. The backbone of any software engineering process is the description of *who* (the role of the actor of a process) does *what* (artifacts) and *how* (activities). Thus each of these specification languages builds ontology of processes by applying concepts such as *activity* and the *actor* of the activity. Some of them take into account also *artefacts* on which a process operates (BPMN, PSL, RUP). From the logico-linguistic point of view, each process can be explicated as an event represented as an *activity*. Activities are denoted by verbs, and *actor* and *artefacts* are the so called verb-valency participants. However, since there are frequently more kinds of participants than just an *actor* and *artefacts*, a more detailed analysis of event participants is called for. Consider for example the event *John is going to Ostrava by train at the speed 100 km/h*. The action *going* has the following participants: actor (*John*), instrument of transport (*train*) and measure of its execution (*100 km/h*). The analysis of events with such a more complex structure is particularly important when building up a multi-agent system (MAS). For these reasons we are going to propose a detailed analysis of participants of an event.

The starting point of an ontological analysis of processes is the fact that each process can be specified by a verb (*what* is to be done). Thus it seems to be appropriate to make use of the results of linguistic analysis of verbs, to wit of the theory of *verb-valency frames*, which is one of the goals of this paper. According to this theory each verb is inherently connected with the so-called *participants*. They are parameters of an action or process denoted by the verb, like the agent/actor of the process (*who*), the objects that the process operates on, resources of the process, etc. Thus verb-valency frames roughly correspond to senses of verbs, and by their exploitation we can obtain a fine-grained specification of processes. There are valency dictionaries (VALLEX, VerbaLex) which provide classifications of verb participants, and I am going to make use of these resources. As another source of information I will explore Sowa's case relations of his conceptual graphs. Based on the analysis and comparison of these three resources, I propose typical process ontology together with the most important types of participants assigned to processes as their requisites. As for terminology, I will use the terms 'process' and 'event' almost interchangeably. This is due to the fact that each (abstract) process is a structured entity that consists of constituents. But constituents of a process are not its products which are beyond the process; rather, particular constituents are again (lower-level) processes. At the lowest level of abstraction there are constituent processes which are not further decomposed, which I will call 'events'.

My background theory is the *procedural semantic* framework of Transparent Intensional Logic (TIL). The basic notion of TIL is a *construction*. Constructions are abstract *procedures* (or processes) assigned to expressions as their context-invariant structured meaning. TIL operates smoothly at three levels of abstraction, namely hyperintensional, intensional and extensional level. Hyperintensional level is the realm of conceptual objects viewed as structured procedures, while intensional and

---

<sup>2</sup> For details see [3].

<sup>3</sup> For details see [4].

<sup>4</sup> PSL ontology is a formal set of axioms and theories as in FOL. There are 17 axioms defining this theory and one can find summary of them in [http://www.mel.nist.gov/psl/psl-ontology/psl\\_core.html](http://www.mel.nist.gov/psl/psl-ontology/psl_core.html).

<sup>5</sup> For details see [5].

extensional level is the realm of unstructured set-theoretical functions. At the intensional level the object of predication is the whole function while at the extensional level the value of the function. Hence TIL is a semantically expressive framework apt for the conceptual analysis of processes that makes it possible to properly differentiate between processes and their products, as well as to classify the products in a fine-grained way. Thus we voted for TIL is a specification language for modelling ontology of processes viewed as an extensional *logic of (hyper-)intensions*.<sup>6</sup>

The rest of the paper is organised as follows. Section 2 is a brief introduction to the theory of verb-valency frames. In Section 3, three classifications of participants according to the linguistic dictionaries VALLEX, VerbaLex and Sowa's thematic roles are analysed. TIL is introduced in Section 4, and in Section 5 requisites and typical properties are defined using TIL. These relations are assigned to events in order to make their ontology more precise. Based on the analysis of verb-valency frames, an event ontology is proposed in Section 6. This proposal is illustrated by an example from the area of multi-agent systems.

## 2. The theory of verb-valency frames

In [7, p. 111] Buitelaar, Cimiano and Hasase refer about our needs of linguistic establishing of ontology as follows: "...a grounding in natural language is needed for several reasons:

- When engineering an ontology, human developers will be able to better understand and manipulate ontologies. Associating linguistic information to ontologies (in the simplest form by labels) allows people to ground concepts and relations defined in the ontology with their own linguistic and cognitive systems.
- In ontology population, automatic procedures for ontology-based information extraction from text will be better equipped to link textual data with ontology elements when these are associated with information on their linguistic realization.
- In verbalizing an ontology, i.e., in generating natural language text descriptions, richer models that capture how concepts and relations are realized linguistically will be needed."

Since the goal of this paper is to propose a method of building up ontology of events founded linguistically, we are first going to deal with the theory of verb-valency frames. As mentioned above, each event can be specified by a verb, and the semantics of the respective verb is provided via its valency frame. In general, valency is the ability of a verb (or another word class) to bind other formal units, i.e. words, which cooperate to provide its meaning completely. These units are so called *functors* or *participants*. Thus valency of a verb determines the number of arguments (participants) controlled by a verbal predicate. This ability of verbs (or lexeme in general) results from their meaning rather than from formal aspects. Despite this fact, lexical and grammatical valency are distinguished. *Grammatical valency* contains information about formal aspects of a verb, such as grammatical case, and *lexical valency* contains information about the semantic character of particular participants. Grammatical valency depends on a language that is being analysed. On the other hand, lexical valency is language independent, since it is established semantically. Hence we have to

---

<sup>6</sup> In [6] we also studied and proposed a domain ontology based on TIL. As mentioned above, in this paper I concentrate on a process ontology.



pay attention to the lexical valency in order to build up an event ontology independent of a used language. More details on the theory of valency frames can be found, for instance, in [8] and [9].

Valency participants can play an obligatory or a facultative role. Consider, for example, the verb *chastise*. This verb has two obligatory participants *who* (agent) and *whom* (patient). Moreover, this verb can be connected with other facultative participants which express inter alia locality and time such as in the following sentence: *A teacher chastises a student in the school early in the morning*. It would be useful to classify verb participants into types according to their semantics. Yet there are many classifications of participant types described in the literature. In the next section we are going to deal with some of them.

### 3. Classifications of participants

To provide a detailed specification of particular classifications of participants is out of the scope of this paper. Their thorough comparison according to the valency dictionaries (VALLEX, VerbaLex) and Sowa's case relations can be found in [10].

The electronic version of the valency dictionary VALLEX has been developed since 2001 in the Institute of Formal and Applied Linguistic, Faculty of Mathematics and Physics, Charles University of Prague, and the electronic version of the valency dictionary VerbaLex has been developed in the Natural Language Processing Centre FI MU in Masaryk University of Brno.<sup>7</sup> These dictionaries provide the list of lexical and grammatical valency frames.

John Sowa uses for the valency participants the term *thematic roles* or *case relations* and he represents them by *conceptual graphs*. His summary of all the thematic roles can be found in [1] or in the electronic source *Thematic roles* [13]. In the conceptual graphs thematic roles are represented via their conceptual relations. These relations link the concept represented by a verb to the concepts of its participants. Here is an example of a conceptual graph taken from [1, p.505]:

*The ambulance arrived quickly.*

[Ambulance:#] ← (Thme) ← [Arrive] → (Manr) → [Quick].

Paraphrase: *The arrival of the ambulance had a quick manner.*"

Now these three sources, that is VALLEX, VerbaLex and Sowa's conceptual graphs, will be compared in order to propose the list of participants applicable to and suitable for the event ontology. The main problem that complicated the comparison is the fact that the authors of these ontologies mix up two different viewpoints of classifying, namely a functional and semantic aspect.

As an example, let us compare the categories of participants ART(ifact), OBJ(ect), ACT(ivity), EVEN(t) with the categories AG(ens), ATTR(tribute), CAUSE, PAT(ient). Names of the participants of the first group result from the effort to capture some ontological essence. From the conceptual point of view, it is a classification of entities on the grounds of their membership into some entity sort. On the other hand, names of the participants in the second group such as AG, ATTR, CAUSE and PATient express

---

<sup>7</sup> For details on the Prague dictionary VALLEX see [11], and the table of the main semantic participants concerning lexical valency according to VerbaLex can be found in [12].

their role in the activity referred to by the respective verb. For the sake of simplicity we shall refer to these two approaches to the specification of participants as the *semantic* and *functional* approach, respectively. Somewhere in between these two groups there are participants which provide location, measure and time specifications.

For illustration, we now apply these functional, semantic and location/measure/time characteristics into the classification of participants according to VerbaLex. The table below contains VerbaLex roles. We mark the additional aspects as follows. Functional roles are bold printed, semantic roles are in the normal font and location, measure and time participants are italicised.

Table 1: Classification of VerbaLex participants according to the functional and semantic approach.

<b><i>Abbr.</i></b>	<b><i>Definition</i></b>
ABS	abstraction:1 - a concept or idea not associated with any specific instance
ACT	act:2 - something that people do or cause to happen
<b>AG</b>	the semantic role of the animate entity that instigates or causes the happening denoted by the verb in the clause - not from EWN Top-Ontology
ANY	anything:1 - a thing of any kind
ART	artifact:1 - a man-made object taken as a whole
<b>ATTR</b>	attribute:2 - an abstraction belonging to or characteristic of an entity
<b>CAUSE</b>	cause:4 - any entity that causes events to happen
COM	communication:2 - something that is communicated by or to or between people
ENT	entity:1 - that which is perceived or known or inferred to have its own physical existence (living or nonliving)
EVEN	event:1 - something that happens at a given place and time
<i>EXT</i>	extent:2 - the distance or area or volume over which something extends
FEEL	feeling:1 - the psychological feature of experiencing affective and emotional states
GROUP	group:1 - any number of entities (members) considered as a unit
INFO	info:1 - a message received and understood that reduces the recipient's uncertainty
<b>INS</b>	instrument:1 - a device that requires skill for proper use
KNOW	knowledge:1 - the psychological result of perception and learning and reasoning
<i>LOC</i>	location:1 - a point or extent in space
<b>MAN</b>	manner:1 - a manner of performance
OBJ	object:1 - a tangible and visible entity; an entity that can cast a shadow
<b>PART</b>	part:1 - something determined in relation to something that includes it
<b>PAT</b>	patient:2 - the semantic role of an entity that is not the agent but is directly involved in or affected by the happening denoted by the verb in the clause - not from EWN Top-Ontology
PHEN	phenomenon:1 - any state or process known through the senses rather than by intuition or reasoning
<b>POS</b>	possession:2 - anything owned or possessed
<b>REAS</b>	reason:1 - a rational motive for a belief or action
<b>REC</b>	recipient:1 - a person who gets something
<b>SOC</b>	associate:1 - a person who joins with others in some activity
STATE	state:4 - the way something is with respect to its main attributes
SUBS	substance:1 - that which has mass and occupies space
<i>TIME</i>	time:5 - the continuum of experience in which events pass from the future through the present to the past

From the viewpoint of building ontology of *events* it is appropriate to classify participants according to their functionality specified by the verbal predicate rather than their semantic character. The role a participant plays in the is primary for the ontology of events while the semantic character is secondary. This functional aspect of participants is determined by the meaning of the verb while their semantic aspect is determined by the membership of an individual that plays the role of the participant into a descriptive or entity sort. Moreover, objects of a different semantic character can play one and the same functional role in the given process.

As mentioned above, the goal of this paper is to propose an event ontology based on the comparison of the above three linguistic classifications. A thorough comparison between these classifications of verb participants is out of the scope of this paper, and the details can be found in [10]. The crucial point was to differentiate between the functional and semantic aspect of the classification. Then the functional types were considered, and since the linguistic classification is too detailed, our goal was to choose the most important types of participants. As a result, we obtained this classification applicable to and suitable for the event ontology.

PAT - patiens

ADR – addressee

BEN - beneficent (somebody who has a benefit from an event)

MAN – manner of event execution (measure, speed etc.)

INST – instrument

DIR1 – direction of event – *from where*

DIR2 - direction of event – *which way*

DIR3 - direction of event – *where to*

Particular participants of an event can have facultative or obligatory character according to the character of the event. Obligatory participants are specified as *requisites* and facultative participants as *typical properties* of an event. Here just briefly.  $P$  is a requisite of an event  $E$  iff necessarily, whenever and wherever  $E$  obtains  $P$  is the case as well. For instance, if an agent  $a$  is driving then there must be a vehicle used in  $a$ 's driving. Similarly,  $Q$  is a typical property of  $E$  iff typically, whenever and wherever  $E$  obtains  $P$  is the case as well, unless there is an exception. For instance, we can specify that driving a car is a typical property assigned to the event of  $a$ 's going from Prague to Brno. Hence  $a$  is typically driving when going from Prague to Brno unless there is an exception, for instance  $a$  takes a train.

Now we are going to put these considerations on a more solid ground. In [6] we introduced our approach to ontology building and characterized ontology as a logic of intensions, that is the logic that examines properties of and relations between higher-order entities like concepts. To this end we applied Transparent Intensional Logic (TIL) which is our background theory in this paper as well. TIL operates with a single *procedural semantics* for all kinds of logical-semantic context, whether extensional, intensional or hyper-intensional, while adhering to the compositionality principle throughout. Moreover, it is within the capacity of TIL to explicitly distinguish analytical vs. empirical knowledge and to specify particular degrees of necessity. Thus we will make use of these TIL features to distinguish between facultative and obligatory character of participants.

#### 4. TIL in brief

TIL is a logical system apt for logical analysis of natural language. It was developed by Pavel Tichý who introduced the main principles of TIL in [14]. The complete works from Tichý can be found in [15]. TIL has further been developed for instance by M. Duží, P. Materna and B. Jespersen and a great deal of the TIL contemporary research can be found in [16]. The outcomes of this book are especially relevant to the inter alia conceptual modeling area. In the next paragraphs I draw in particular on material from [17, chapter 2].

In modern jargon, TIL belongs to the paradigm of structured meanings. Since the late sixties many logicians and semanticists aimed at hyperintensionally explicated meaning, because it became obvious that intensional or possible-world semantics is too coarse-grained. Various adjustments of Frege's semantic schema have been proposed, shifting the entity named by an expression from the extensional level of atomic (physical/abstract) objects to the intensional level of molecular objects such as sets or functions/mappings. Yet natural language is rich enough to generate expressions that talk neither about extensional nor intensional objects. Propositional attitudes are notoriously known as the hard cases that are neither extensional nor intensional, as Carnap in [18] characterized them. It has become increasingly clear since the 1970s that we need to individuate meanings more finely than by possible-world intensions, and the need for *hyperintensional* semantics is now broadly recognised. Our position is a plea for hyperintensional semantics, which takes expressions as encoding *algorithmically structured procedures* producing extensional/intensional entities (or lower-order procedures) as their products. This approach — which could be characterized as being informed by an *algorithmic* or *computational turn* — has been advocated by, for instance, Moschovakis in [19]. Yet much earlier, in the early 1970s, Tichý introduced his notion of *construction* and developed the system of Transparent Intensional Logic (TIL).<sup>8</sup>

Constructions, as well as the entities they construct, all receive a type. The ontology of TIL is organized in an infinite, bi-dimensional hierarchy of types. Since we strictly distinguish between a construction of an object and the object itself, and between a function and its value, construction must be always of a higher order than the object it constructs, and a function is of a higher degree than its value. Thus one dimension of the type hierarchy increases molecular complexity of functions, the other dimension increases the order of constructions. Our definitions are inductive, and they proceed in three stages. First, we define the simple types of order 1 comprising non-constructions. Then we define constructions and, finally, the ramified hierarchy of types.

**Definition 1** (*Types of order 1*) Let  $B$  be a base, i.e., a collection of non-empty sets.

- i) Every member of  $B$  is a *type of order 1*.
- ii) Let  $\alpha, \beta_1, \dots, \beta_m$  be *types of order 1*. Then the set  $(\alpha\beta_1 \dots \beta_m)$  of partial functions with values in  $\alpha$  and arguments in  $\beta_1, \dots, \beta_m$ , respectively, is a *type of order 1*.
- iii) *Nothing is a type of order 1* unless it follows from i) and ii).

---

<sup>8</sup> See Tichý [14] and [15].

The types *ad* (ii) are *functional* types. They are sets of *partial functions*, i.e., functions that associate every *m*-tuple of arguments with *at most* one value. Thus total functions are a special kind of partial functions.

The choice of the base depends on the area and language we happen to be investigating. When investigating purely mathematical language, the base can consist of, e.g., two atomic types; *o*, the type of truth-values, and *v*, the type of natural numbers. When analyzing an ordinary natural language, we use the *epistemic base* which is a collection of four atomic types,  $\{o, \iota, \tau, \omega\}$ , where

- $o = \{\mathbf{T}, \mathbf{F}\}$ , the set of *truth-values*,
- $\iota$  = the universe of discourse (members: *individuals*),
- $\tau$  = the set of *real numbers* (or of *time moments*),
- $\omega$  = the logical space, the set of *possible worlds*.

Since *function* rather than *relation* is a primitive notion of TIL, we model *sets* and *relations* by their characteristic functions. Thus, for example, the set of prime numbers is a function of type  $(o\tau)$  that associates any number with **T** or **F** according as the given number is a prime.

**Definition 2** (*intension* and *extension*): (*PWS*) *intensions* are entities of type  $(\beta\omega)$ : mappings from possible worlds to some type  $\beta$ . The type  $\beta$  is frequently the type of the *chronology* of  $\alpha$ -objects, i.e., mapping of type  $(\alpha\tau)$ . Thus  $\alpha$ -intensions are frequently functions of type  $((\alpha\tau)\omega)$ , abbreviated as ' $\alpha_{\tau\omega}$ '. *Extensions* are entities of a type  $\alpha$  where  $\alpha \neq (\beta\omega)$  for any type  $\beta$ .

*Examples* of frequently used intensions are:

- *Propositions* (denoted by declarative sentences) are of type  $o_{\tau\omega}$ ;
- *properties of individuals* (usually denoted by nouns or intransitive verbs like 'is a student', 'walks') are of type  $(o\iota)_{\tau\omega}$ ;
- *binary relations-in-intension* between individuals are of type  $(o\iota\iota)_{\tau\omega}$ ;
- *individual offices/roles* (cf. Church's individual concepts, usually denoted either by superlatives like 'the highest mountain' or terms with built-in uniqueness, like 'The President of the USA') are of type  $\iota_{\tau\omega}$ .

*Expressions* which denote non-constant intensions (i.e. functions that take different values in at least two world-time pairs) are *empirical*. Note that some extensions involve the set of possible worlds, but not as their domain. For instance, a *set* of propositions is an extensional entity of type  $(oo_{\tau\omega})$ . On the other hand, a *property* of propositions, like being true in a world *w* at time *t*, is an intensional entity of type  $(oo_{\tau\omega})_{\tau\omega}$ .

*Quantifiers*  $\forall^\alpha, \exists^\alpha$  are extensions, viz. type-theoretically polymorphous functions of type(s)  $(o(o\alpha))$  defined as follows: The *universal quantifier*  $\forall^\alpha$  is a function that associates a class *C* of  $\alpha$ -elements with **T** if *C* contains all elements of the type  $\alpha$ , otherwise with **F**. The *existential quantifier*  $\exists^\alpha$  is a function that associates a class *C* of  $\alpha$ -elements with **T** if *C* is a non-empty class, otherwise with **F**.

The *singulariser*  $Sing^\alpha$  is a partial type-theoretically polymorphic function of type(s)  $(\alpha(o\alpha))$  that associates a class *C* with the only  $\alpha$ -element of *C* if *C* is a singleton, otherwise the function  $Sing^\alpha$  is undefined.

Where  $A$   $v$ -constructs a truth-value, i.e. an  $o$ -object and  $x$   $v$ -constructs an  $\alpha$ -object, we will often use the abbreviated notation ‘ $\forall x A$ ’, ‘ $\exists x A$ ’ and ‘ $\iota x A$ ’ instead of ‘ $[^0\forall^\alpha \lambda x A]$ ’, ‘ $[^0\exists^\alpha \lambda x A]$ ’, ‘ $[^0\text{Sing}^\alpha \lambda x A]$ ’, respectively, when no confusion can arise.

Constructions are assigned to expressions as their algorithmically structured, context-invariant meanings. When claiming that constructions are algorithmically structured, we mean the following. A construction  $C$  consists of one or more particular steps, or *constituents*, that are to be individually executed in order to execute  $C$ . The objects a construction operates on are not constituents of the construction. Just like the constituents of a computer program are its sub-programs, so the constituents of a construction are its sub-constructions. Thus on the lowest level of non-constructions, the objects that constructions work on have to be supplied by other (albeit trivial) constructions.

Constructions themselves may occur not only as constituents to be executed, but also as objects that still other constructions operate on. Therefore, one should not conflate *using* constructions as constituents of compound constructions and *mentioning* constructions that enter as input/output objects into compound constructions. So we must strictly distinguish between *using* constructions as constituents and *mentioning* constructions as objects.

Mentioning is, in principle, achieved by *using* atomic constructions. A construction  $C$  is *atomic* if it does not contain any other construction as a used sub-construction (a ‘constituent of  $C$ ’) but  $C$ . There are two atomic constructions that supply entities (of any type) on which compound constructions operate: *Variables* and *Trivializations*. *Compound* constructions, which consist of other constituents than just themselves, are *Composition* and *Closure*. *Composition* is the instruction to apply a function to an argument in order to obtain its value (if any) at the argument. It is *improper*, i.e., does not construct anything, if the function is not defined at the argument. *Closure* is the instruction to construct a function by abstracting over variables in the ordinary manner of  $\lambda$ -calculi. Finally, higher-order constructions can be used once or twice over as constituents of constructions. This is achieved by a fifth and sixth construction called *Execution* and *Double Execution*, respectively.

### Definition 3 (construction)

- i) The *Variable*  $x$  is a **construction** that constructs an object  $O$  of the respective type dependently on a valuation  $v$ ; it  $v$ -constructs  $O$ .
- ii) *Trivialization*: Where  $X$  is an object whatsoever (an extension, an intension or a *construction*),  ${}^0X$  is the **construction** *Trivialization*. It constructs  $X$  without any change.
- iii) The *Composition*  $[X Y_1 \dots Y_m]$  is the following **construction**. If  $X$   $v$ -constructs a function  $f$  of a type  $(\alpha \beta_1 \dots \beta_m)$ , and  $Y_1, \dots, Y_m$   $v$ -construct entities  $B_1, \dots, B_m$  of types  $\beta_1, \dots, \beta_m$ , respectively, then the *Composition*  $[X Y_1 \dots Y_m]$   $v$ -constructs the value (an entity, if any, of type  $\alpha$ ) of  $f$  on the tuple-argument  $\langle B_1, \dots, B_m \rangle$ . Otherwise the *Composition*  $[X Y_1 \dots Y_m]$  does not  $v$ -construct anything and so is *v-improper*.
- iv) The *Closure*  $[\lambda x_1 \dots x_m Y]$  is the following **construction**. Let  $x_1, x_2, \dots, x_m$  be pairwise distinct variables  $v$ -constructing entities of types  $\beta_1, \dots, \beta_m$  and  $Y$  a construction  $v$ -constructing an  $\alpha$ -entity. Then  $[\lambda x_1 \dots x_m Y]$  is the **construction**  *$\lambda$ -Closure* (or *Closure*). It  $v$ -constructs the following function  $f/(\alpha \beta_1 \dots \beta_m)$ . Let  $v(B_1/x_1, \dots, B_m/x_m)$  be a valuation identical with  $v$  at least up to assigning objects

$B_1/\beta_1, \dots, B_m/\beta_m$  to variables  $x_1, \dots, x_m$ . If  $Y$  is  $v(B_1/x_1, \dots, B_m/x_m)$ -improper (see iii), then  $f$  is undefined on  $\langle B_1, \dots, B_m \rangle$ . Otherwise the value of  $f$  on  $\langle B_1, \dots, B_m \rangle$  is the  $\alpha$ -entity  $v(B_1/x_1, \dots, B_m/x_m)$ -constructed by  $Y$ .

- v) The *Execution*  ${}^1X$  is the **construction** that either  $v$ -constructs the entity  $v$ -constructed by  $X$  or, if  $X$   $v$ -constructs nothing, is  $v$ -improper.
- vi) The *Double Execution*  ${}^2X$  is the following **construction**. Let  $X$  be any entity; the *Double Execution*  ${}^2X$  is  $v$ -improper (yielding nothing relative to  $v$ ) if  $X$  is not itself a construction, or if  $X$  does not  $v$ -construct a construction, or if  $X$   $v$ -constructs a  $v$ -improper construction. Otherwise, let  $X$   $v$ -construct a construction  $X'$  and  $X'$   $v$ -construct an entity  $Y$ . Then  ${}^2X$   $v$ -constructs  $Y$ .
- vii) Nothing is a **construction**, unless it so follows from (i) through (vi).  $\square$

*Notation and abbreviations.*

- ‘ $X/\alpha$ ’ means that the object  $X$  is (a member) of type  $\alpha$ ;
- ‘ $X \rightarrow_v \alpha$ ’ means that the type of the object  $v$ -constructed by  $X$  is  $\alpha$ . We use ‘ $X \rightarrow \alpha$ ’ if what is  $v$ -constructed does not depend on a valuation  $v$ .
- We will standardly use the variables  $w \rightarrow_v \omega$  and  $t \rightarrow_v \tau$ ;
- If  $C \rightarrow_v \alpha_{\tau\omega}$ , the frequently used Composition  $[[C w] t]$ , the intensional descent of the  $\alpha$ -intension  $v$ -constructed by  $C$ , will be written as ‘ $C_{wt}$ ’.
- When using constructions of truth-value functions, namely  $\wedge$  (conjunction),  $\vee$  (disjunction) and  $\supset$  (implication) of type (ooo), and  $\neg$  (negation) of type (oo), we often omit Trivialisation and use infix notation.
- When using identity relations  $=^{\alpha}/(o\alpha\alpha)$ , we often omit the superscript  $\alpha$  and use infix notation, whenever no confusion arises.

As mentioned above, constructions themselves are objects and thus also receive a type. Only it cannot be a type of order 1, because a construction cannot be of the same type as the object it constructs. Constructions that construct entities of order 1 are *constructions of order 1*. They belong to a *type of order 2*, denoted by ‘ $*_1$ ’. This type  $*_1$ , together with atomic types of order 1, serves as the base for the following induction rule: any collection of partial mappings, type  $(\alpha \beta_1 \dots \beta_n)$ , involving  $*_1$  in their domain or range is a *type of order 2*. Constructions belonging to the type  $*_2$ , which identify entities of order 1 or 2, and partial mappings involving such constructions, belong to a *type of order 3*; and so on *ad infinitum*.

The definition of the ramified hierarchy of types decomposes into three parts. First, simple types of order 1 were already defined by Definition 1. Second, we define constructions of order  $n$ , and third, types of order  $n + 1$ .

**Definition 4 (Ramified hierarchy of types)**

**$T_1$  (types of order 1).** See Definition 1.

**$C_n$  (constructions of order  $n$ )**

- i) Let  $x$  be a variable ranging over a type of order  $n$ . Then  $x$  is a *construction of order  $n$  over  $B$* .
- ii) Let  $X$  be a member of a type of order  $n$ . Then  ${}^0X$ ,  ${}^1X$  and  ${}^2X$  are *constructions of order  $n$  over  $B$* .
- iii) Let  $X, X_1, \dots, X_m$  ( $m > 0$ ) be constructions of order  $n$  over  $B$ . Then  $[X X_1 \dots X_m]$  is a *construction of order  $n$  over  $B$* .

- iv) Let  $x_1, \dots, x_m, X$  ( $m > 0$ ) be constructions of order  $n$  over  $B$ . Then  $[\lambda x_1 \dots x_m X]$  is a construction of order  $n$  over  $B$ .
- v) Nothing is a construction of order  $n$  over  $B$  unless it so follows from  $\mathbf{C}_n$  (i)-(iv).

### $\mathbf{T}_{n+1}$ (types of order $n + 1$ )

Let  $*_n$  be the collection of all constructions of order  $n$  over  $B$ .

- i)  $*_n$  and every type of order  $n$  are types of order  $n + 1$ .
- ii) If  $0 < m$  and  $\alpha, \beta_1, \dots, \beta_m$  are types of order  $n + 1$  over  $B$ , then  $(\alpha \beta_1 \dots \beta_m)$  (see  $\mathbf{T}_1$  ii) is a type of order  $n + 1$  over  $B$ .
- iii) Nothing is a type of order  $n + 1$  over  $B$  unless it so follows from  $\mathbf{T}_{n+1}$  (i) and (ii).  $\square$

So much for the logical machinery we need to specify ontology of events.

## 5. Requisites and typical properties

The process of an ontology design usually begins with the specification of primitive concepts, i.e., Trivializations of objects that are not constructions.<sup>9</sup> These primitive concepts are supposed to be commonly understood and they are not further refined. For instance, primitive concepts of a traffic-system ontology might be  ${}^0Agent$ ,  ${}^0Lane$ ,  ${}^0Crossroads$ , etc. Next we specify compound concepts as ontological definitions of entities of a given domain. For instance, a road can be defined as consisting of two or more lanes which pass from a crossroad to another crossroad.

When specifying relations between entities, we have to distinguish between empirical and analytical relations. The former are relations-in-intension, mostly between individuals, like the part-whole relation. Analytical relations are relations-in-extension between intensions, for instance, a requisite relation and a property typical for another property. These relations give rise to ISA taxonomies. For instance, that a driver is a person is analytically necessary proposition  $TRUE$  that takes the value  $\mathbf{T}$  in all  $\langle w, t \rangle$ -pairs. The requisite relation of type  $(o(o\iota)_{\tau\omega})(o\iota)_{\tau\omega}$  between the property of being a driver and the property of being a person is defined as follows:

$$[{}^0Req \ {}^0Person \ {}^0Driver] =_{df} \forall w \forall t [\forall x [[{}^0Driver_{wt} x] \supset [{}^0Person_{wt} x]]]$$

*Gloss.* Being a person is a requisite of being a driver. In other words, necessarily and for any individual  $x$ , if  $x$  instantiates the property of being a driver then  $x$  also instantiates the property of being a person.

On the other hand, a driver typically owns a car, unless he is a chauffeur or a chauffeuse working for somebody else without owning a car. We say that owning a car is a typical property of a driver:

$$[{}^0Typically \ {}^0Own\_Car \ {}^0Driver \ {}^0Exception] =_{df} \forall w \forall t [\forall x \neg [{}^0Exception_{wt} x] \supset [[{}^0Driver_{wt} x] \supset [{}^0Own\_Car_{wt} x]]]$$

Requisites and typical properties can obtain between intensions of any type. Here we define these relations only between properties of individuals. The other kinds can be easily deduced from this one. Let  $p, q, exc \rightarrow_v (o\iota)_{\tau\omega}$ ;  $x \rightarrow_v \iota$ ;  $True/(oo_{\tau\omega})_{\tau\omega}$ : the

<sup>9</sup> In the next paragraphs I draw on material from [17, chapter 3].



property of a proposition of being true in a world  $w$  at time  $t$ . Then  $q$  is a *requisite* of  $p$ , if and only if

$$\forall w \forall t \forall x [[{}^0True_{wt} \lambda w \lambda t [p_{wt} x]] \supset [{}^0True_{wt} \lambda w \lambda t [q_{wt} x]]]$$

The relation of being a typical property is defined as follows:

$$\forall w \forall t \forall x [\neg [{}^0True_{wt} \lambda w \lambda t [exc_{wt} x]] \supset [{}^0True_{wt} \lambda w \lambda t [p_{wt} x]] \supset [{}^0True_{wt} \lambda w \lambda t [q_{wt} x]]]$$

*Gloss:*  $p$  is typical of  $q$  unless *exc(ption)*.

*Note.* Due to partiality, we must use here the property of propositions of being true. It returns the value **T** if the given proposition takes the value **T** in a  $\langle w, t \rangle$  pair, otherwise **F**. If we did not apply this property, then it might be the case that  $[p_{wt} x]$  would be  $\nu$ -improper and thus the whole Composition  $[[{}^0True_{wt} \lambda w \lambda t [p_{wt} x]] \supset [{}^0True_{wt} \lambda w \lambda t [q_{wt} x]]]$  would be  $\nu$ -improper as well, which means that the above Closure would construct **F**. This is wrong, for sure, because the relation of being a requisite, providing it is valid, then it is valid in all  $\langle w, t \rangle$  pairs.

It is a well-known fact that hierarchies of intensions based on requisite relations establish *inheritance* of attributes and possibly also of operations. For instance, a driver in addition to his/her special attributes like having a driving license inherits all the attributes of a person. This is another reason for including such a hierarchy into ontology.

In the next section we are going to apply the notions of requisite and typical property to the event ontology specification.

## 6. Event ontology specification

Specification of events is driven by the analysis of verbs that denote actions. For instance, the specification of the process of *Charles is driving from Olomouc to Ostrava by train at the speed of 90 km/h* is given by the sense of the verb ‘to drive’ together with its arguments (*who* is driving – the actor, *when* is (s)he driving, *from where*, *to where*, *what kind of vehicle*, etc.). This process thus consists of the set of atomic events such as *Charles turn right/left on the crossroad C1* and so on.

Proposed analysis makes use of Tichý’s [20] where such verbs are called *episodic* verbs. Tichý draws a distinction between episodic and attributive verbs. Episodic verbs (e.g. *drive*, *tell*, etc.) express actions of objects or people as opposed to attributive verbs (e.g. *is heavy*, *looks speedy*) that ascribe some empirical properties. Thus *Charles is driving from Olomouc to Ostrava* is not a condition in which Charles may be. Rather, it is a time consuming *process* consisting of a series of *events*.

In [17] the basic idea of specifying event ontology by means of valency participants treated as requisite and typical properties of events is outlined for the first time. In this paper it is more specific, in particular the type of an action executed within an event is specified.

The verb valency is characterized in linguistics as the ability of a verb to be linked to other terms of the discourse. This ability concerns the semantic level of a language that is the deep structure of a sentence. Since valency frames correspond to senses of verbs, we can obtain a finer specification of the process/procedure. From the logical point of view, a verb denotes a relation and the valency of the verb determines its arity and types of the arguments of the relation. Thus each process can be specified by a verb

(*what* is to be done), possibly with parameters like the agent/actor of the process (*who*), the objects to be operated on, resources, etc. Particular properties of the actor and other parameters are then specified as requisites of the event or its typical properties. Hence, obligatory participants are requisites of the event and facultative participants its typical properties.

Yet before defining the requisite/typical property relation between a process/event and its participants, we must specify the type of an *action* that is to be executed within an event. Such an action is referred to by an episodic verb. Each action has an actor and a set of participants which can be of various types such as individual, property, quantity. Hence an action is a relation between an individual of type  $\iota$  (actor) and participants. Denoting for the sake of simplicity the types of participants  $Part_{a_1}, \dots, Part_{a_n}$  by  $\alpha_1, \dots, \alpha_n$ , respectively, we thus have this type of an action:  $(\circ\iota(\alpha_1 \dots \alpha_n))$ .

As an example, consider the process of *Peter's going to Ostrava by train*. The verb *is going* expresses an action, *Peter* is an actor and *train* and *Ostrava* are participants. The analysis of this process is as follows:

$$\begin{aligned} & \lambda w \lambda t [[{}^0Does_{wt} {}^0Peter {}^0Go] \wedge \\ & \exists x [[{}^0Train_{wt} x] \wedge [{}^0Train = {}^0INST] \wedge [{}^0Train = {}^0Part_{a_i}] \wedge \\ & [{}^0Ostrava = {}^0DIR3] \wedge [{}^0Ostrava = {}^0Part_{a_j}]]] \end{aligned}$$

Notice that *Does* is the relation between an actor and his/her action. Denoting for short the type of an action, i.e.  $(\circ\iota(\alpha_1 \dots \alpha_n))$ , by  $\delta$ , *Does* is of type  $(\circ\iota\delta)_{\tau\omega}$ . Beside *train* and *Ostrava* the process of Peter's going to Ostrava by train can have other participants as well. For this reason we just specify that *train* is one of the participants, say  $i^{th}$ ,  $Part_{a_i}$ , and *Ostrava* another one,  $Part_{a_j}$ .

Definition of the requisite relation between an action and its participants comes down to this:

$$[{}^0Req_a q a] =_{df} \forall w \forall t \forall x [[{}^0Does_{wt} x a] \supset [q = Part_{a_i}]]$$

$$\text{where } q \rightarrow_v \alpha_i; a \rightarrow_v \delta; x \rightarrow_v \iota; Does/(\circ\iota\delta)_{\tau\omega}$$

*Gloss.*  $q$  is a requisite of an action  $a$  iff necessarily ( $\forall w \forall t$ ) for all individuals  $x$  it holds that whenever  $x$  does the action  $a$  then  $q$  is a participant of  $a$ .

For instance, we may want to specify that requisites of the action *Go* are participants of type *INST* (instrument – by what) and *DIR3* (to where). This is done by the following constructions:

$$[{}^0Req_a INST Go] =_{df} \forall w \forall t \forall x [[{}^0Does_{wt} x {}^0Go] \supset \exists y [[y = {}^0INST] \wedge [y = {}^0Part_{a_i}]]]$$

$$[{}^0Req_a DIR3 Go] =_{df} \forall w \forall t \forall x [[{}^0Does_{wt} x {}^0Go] \supset \exists z [[z = {}^0DIR3] \wedge [z = {}^0Part_{a_j}]]]$$

An agent's intention is represented by the object *Want* of type  $(\circ\iota\circ_{\tau\omega})_{\tau\omega}$ . This is a relation between an agent of type  $\iota$  and the proposition (of type  $\circ_{\tau\omega}$ ) which is his/her intention.

Example: *Agent\_A wants to go to Brno.*

$$\begin{aligned} & \lambda w \lambda t [{}^0Want_{wt} {}^0Agent_A \\ & \lambda w \lambda t [[{}^0Does_{wt} {}^0Agent_A {}^0Go] \wedge [{}^0Brno = {}^0DIR3] \wedge [{}^0Brno = {}^0Part_{a_j}]]] \end{aligned}$$

If an agent *B* has in his/her knowledge base all the respective requisite relations of an action *a* and *B* obtains an incomplete message about the action *a* being done by another agent, he/she can ask for the unmentioned participants of the action. Imagine, for instance, that *B* receives a message informing that *A* wants to go to Brno. Hence the content of this message is the above construction. Then *B* might send a query message to *A* asking about the other participants unmentioned in the message, for instance about *INST* (*what vehicle will A go by*), *MAN* (*which speed will A go*) and so on. Hence the explicit specification of requisite relations in agents' knowledge base improves agents' intelligent behaviour.

## 7. Conclusion

We introduced a new linguistically founded approach to event ontology based on the verb-valency frames. Three kinds of classifications of verb participants were compared in order to choose the most important types of participants for event ontology building. We showed that participants can play obligatory or facultative role relative to the meaning of a verbal predicate. Obligatory participants are requisites of the action executed within an event and facultative participants are its typical properties. We specified these requisites and typical properties of actions by means of TIL, and illustrated the need for such a fine-grained ontology by an example of agents' behaviour driven by messaging in a multi-agent system.

**Acknowledgements.** This research has been supported by the project No. CZ.1.07/2.3.00/30.0004 "The enhancement of creation of excellent research teams and intersectoral mobility at Palacky University in Olomouc (POST-UP)". This project is co-financed by the European Social Fund and the state budget of the Czech Republic.

## References

- [1] Sowa, J., F.: *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Brooks, Cole, 2000.
- [2] Workflow management coalition terminology and glossary, *Technical Report, WfMCTC-1011*, Brussels: Workflow Management Coalition, 1996.
- [3] Allweyer, T. (2010): BPMN 2.0 – Introduction to the Standard for Business Process Modeling. BoD. ISBN 978-3-8391-4985-0.
- [4] Liu, P., Zhou, B. (2008): Workflow Mining of More Perspectives of Workflow, In *J. Software Engineering & Applications*, 2008, 1: 83-87.
- [5] Rational Unified Process. Best Practise for Software Development Team. [http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251\\_bestpractices\\_TP026B.pdf](http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf)
- [6] Duží, M., Číhalová, M., Menšík, M.: Ontology as a logic of intensions. In: *Information modelling and Knowledge Base XXII*, Heimbürger A., Kiyoki Y., Tokuda T., Jaakkola H., Yoshida N. (eds.), Amsterdam: IOS Press, 2011, 1, vol. XXII, 1-20.
- [7] Buitelaar, P., Cimiano, P., Haase, P., Sintek, M., Towards Linguistically Grounded Ontologies, *Proceeding ESWC*, Berlin, Springer, 2009, 111-125.
- [8] Lopatková, M., Žabokrtský, Z., Kettnerová, V., *Valenční slovník českých sloves*. Praha, Karolinum, 2008. ISBN 978-80-246-1467-0.
- [9] Fischer, K. and V. Ágel, Dependency grammar and valency theory, *The Oxford handbook of linguistic analysis*, Oxford, Oxford University Press, 2010, 223-255.

- [10] Číhalová, M., *Jazyky pro tvorbu ontologií (Languages for ontology building)*, dissertation work, VŠB-Technical university of Ostrava, Faculty of Electrical Engineering and Computer Science, Department of Computer Science, 2011.
- [11] Lopatková, M., Žabokrtský, Z., Kettnerová, V., *VALLEX 2.5. – Logical structure of the lexicon*, 2006  
URL: [http://ufal.mff.cuni.cz/vallex/2.5/doc/structure\\_en.html#sec:frame](http://ufal.mff.cuni.cz/vallex/2.5/doc/structure_en.html#sec:frame). [10. 1. 2014]
- [12] Pala, K., Hlaváčková, D., *Reprezentace významu sloves (valence a sémantické role)*. *Kognice*, p. 7., Praha, 2010
- [13] Sowa, J. F.: *Thematic roles*.  
URL: <http://www.jfsowa.com/ontology/roles.htm>. [10. 1. 2014]
- [14] Tichý, P., *The Foundations of Frege's Logic*, Berlin, New York, De Gruyter, 1988.
- [15] Tichý, P., *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen (eds.), C., 2004
- [16] Duží, M., Jespersen, B. and Materna, P., *Procedural Semantics for Hyperintensional Logic. Foundations and Applications of Transparent Intensional Logic*. Berlin: Springer, series Logic, Epistemology, and the Unity of Science, vol. 17., 2010.
- [17] Duží, M., Číhalová, M., Menšík, M., Vích, L., *Process ontology*. In sborníku *RASLAN 2010*, eds. Sojka, P., Brno, CNP MUNI, 2011, 77-88. ISBN 978-80-7399-246-0
- [18] Carnap, R., *Meaning and Necessity*, Chicago, Chicago University Press, 1947.
- [19] Moschovakis, Y.N., 'Sense and denotation as algorithm and value', in: J. Väänänen and J. Oikkonen (eds.), *Lecture Notes in Logic*, vol. 2, Berlin, Springer, 1994, 210-49.
- [20] Tichý, P., 'The semantics of episodic verbs', *Theoretical linguistics*, 7, 1980, 263-296. Reprinted in: Tichý (2004, pp. 409-444).

# Musical Tunes Emotions Identification System by means of Intrinsic Musical Characteristics

Tatiana ENDRJUKAITE<sup>a</sup> and Yasushi KIYOKI<sup>b</sup>

<sup>a</sup>*KEIO University, Graduate School of Media and Governance, Kiyoki Laboratory, Tokyo, Japan*

<sup>b</sup>*Prof. of Graduate School of Media and Governance, KEIO University, Tokyo, Japan*

**Abstract.** We design and implement a music-tune analysis system to realize automatic emotion identification and prediction by means of intrinsic musical features. To compute physical elements of music pieces we define three significant tunes parameters. These are: repeated parts or repetitions inside a tune, thumbnail of a music piece, and homogeneity pattern of a tune. They are significant, because they are related to how people perceive music pieces. By means of these three parameters we can express the essential features of emotional-aspects of each piece. Our system consists of music-tune features database and computational mechanism for comparison between different tunes. Based on Hevner's emotions adjectives groups we created a way of emotion presentation on emotion's plane with two axes: activity and happiness. That makes it possible to determine perceived emotions of listening to a tune and calculate adjacent emotions on a plane. Finally, we performed a set of experiments on western classical and popular music pieces, which presented that our proposed approach reached 72% precision ratio and show a positive trend of system's efficiency when database size is increasing.

**Keywords:** music, emotions, repetitions, tune's thumbnail, tune's internal homogeneity.

## 1. Introduction

Huge amount of music pieces are generated every day all over the world, making it very difficult to find desired ones among them. Descriptions of music pieces such as genre, artist, tempo, style and others help to decide if it is what we are searching for or not. However, to find a new tune we will like, we have to listen to each piece to find it out. This could be very time consuming. In addition, there are various musical features that influence people as they listen, and after they listen to music [9].

It is well established that human beings respond emotionally to music, little is known about precisely what it is in the music that they are responding to [6]. In recent years the design and implementation of tunes playlist suggestion systems is one of the key issues in the field of multimedia research. In the design of such system the important issue is how to define and represent the internal features of music pieces and how to select tunes according to the user's impression.

In this paper we focus on dependency between intrinsic characteristics of tunes and emotions people experience while listening to those tunes. By analyzing repetitions and

tunes internal homogeneity we may discover why a piece of music influences us the way it does. We try to predict what influence an unknown tune would have on us before we listen to it.

The ultimate goal of this research is to construct a system for determining the expected emotional effect of listening to a tune and to determine mapping between music characteristics and emotions. Such system could be used in tunes playlist suggestion tasks. For example, the proposed method can be used in a system designed for selecting appropriate tunes according to the user's wishes to suggest these tunes for listening to the user.

However, there are people with very different views on music, but we cannot say there is a total chaos in music perception by people. It is well known that there is a notable relation between some types of music and emotions which people experience on them. It is also important to keep in mind that we can split all people into groups with similar perception of music, or even consider only one person individually to have a personalized emotion detection system.

This paper is organized as follows: Section 2 covers related work. Section 3 describes features extracted from tunes and techniques for processing them to add music pieces to database. Tunes descriptors creation approach is presented in Section 4. Section 5 presents the way the system calculates tunes emotions step by step. Results analysis and discussions represented in section 6. Conclusions and future work are covered in Section 7.

## **2. Related Work**

At present, there is no complete theory for automatically analyzing music structure. However, there are encouraging researches on music signal processing and retrieval detecting the most frequently appearing component in a piece of music based on music structure analysis. Generally speaking, the more repetitions and similar phases there are in a piece of music, the easier it is for people to have affinity for it.

The main motivation of this research is to determine music characteristics mapping to emotions. Regarding that motivation, researches have made genre recognition analysis [11, 12] and music similarity analysis by detecting the most representative part of a piece of music [2, 3]. The most representative part in a piece has been defined as the most frequently repeated component in it such as repetitions and typical parts during tune's performance. In particular, PCM acoustic data has been used. Repetitions have found by performing self-similarity calculations with Mel-Frequency Cepstral Coefficients (MFCCs) [4]. Tune's internal homogeneity automatic identification by typical parts within tune have been detected by performing instantaneous frequency spectrum (IFS) [5].

The relation between musical sounds and their influence on the listener's emotion were studied by Hevner through experiments, which substantiated a hypothesis that music inherently carries emotional meaning [1]. We want to find information extractable directly from music that, after appropriate processing, will help people to find new and unknown tunes they likely would like to listen to. The information we seek is mainly discussed in acoustic psychology [7, 10, 13] and as such, its significance is an on-going research topic. Accordingly, this research should be based on music psychology and at the same time this research should contribute to musical research.

### 3. Music System creation

The system of music processing to detect tune's emotion is outlined in figure 1, where white boxes show data and blue boxes show processing steps. Acoustic data is used as an input. Output is a detected tune's emotion. Tunes database store information about every tune and its corresponding emotion that was determined statistically. Tune itself is presented in the database by the tune descriptor, which contains three significant physical parameters of the tunes. Emotion for a tune is stored as a point on the emotional plane (see fig. 3b). This makes it possible to aggregate emotions from multiple tunes into a one single estimation. Explanation of the tune descriptor and the way of emotions aggregation are described below in this paper.

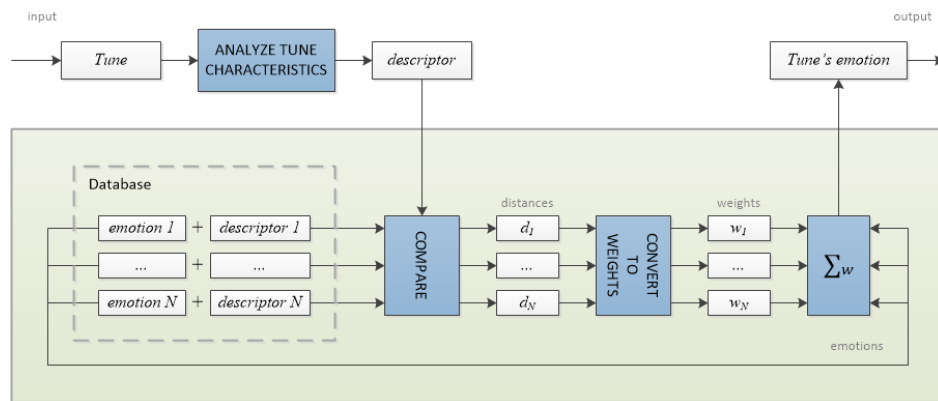


Figure 1. System workflow to detect tune's emotion.

#### 3.1. Tunes emotions data collection

Survey was used for collecting emotions that listeners experience by listening to tunes. An important aspect is that listeners were listening to the full tunes, not only most representative parts. That was required to maximize the precision of emotion estimation. However, well known songs may already have an effect specific to every listener. This happens because there is a strong relation between the tune and the conditions they have been heard before. For example, tune may raise negative emotions if the listener first time heard it in a bad health condition.

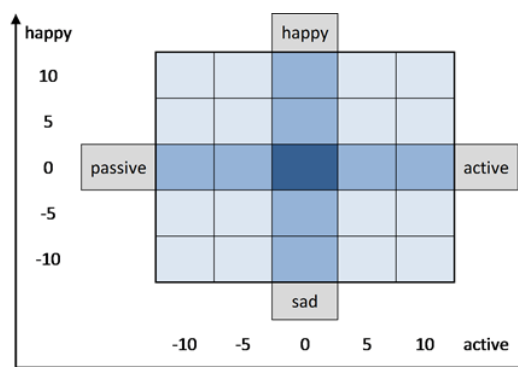
##### 3.1.1. Hevner's system

As a base for the predefined list of emotions we used eight classes of emotions proposed by Kate Hevner in [1]. The Hevner's list of emotions is presented in table 1.

**Table 1.** Hevner's categories

Category	Description
C1	Spiritual, lofty, awe-inspiring, dignified, sacred, solemn, sober, serious
C2	Pathetic, doleful, sad, mournful, tragic, melancholy, frustrated, depressing, gloomy, heavy, dark
C3	Dreamy, yeilding, tender, sentimental, longing, yearning, pleading, plaintive
C4	Lyrical, leisurely, satisfying, serene, tranquil, quiet, soothing
C5	Humorous, playful, whimsical, fanciful, quaint, springtly, delicate, light, graceful
C6	Merry, joyous, gay, happy, cheerful, bright
C7	Exhilarated, soaring, triumphant, dramatic, passionate, sensational, agitated, exciting, impetuous, restless
C8	Vigorous, robust, emphatic, martial, ponderous, majestic, exalting

To make it simpler for listeners to choose the correct emotion we simplified the Hevner's adjective list to a plane with two axes: activeness and happiness from -10 to 10. So, listeners had to put a point on the plane for every listened tune within survey. Compared to the list of 8 classes the plane-approach also has an advantage in results aggregation and further processing. Emotions presentation in a plane is beneficial because it takes into account relation between emotion classes. The given to the listeners emotions plane is presented in the figure 2. For every tune respondents had to put a point on the plane, or select and mark one cell that mostly corresponds to their emotions after listening to the tune.

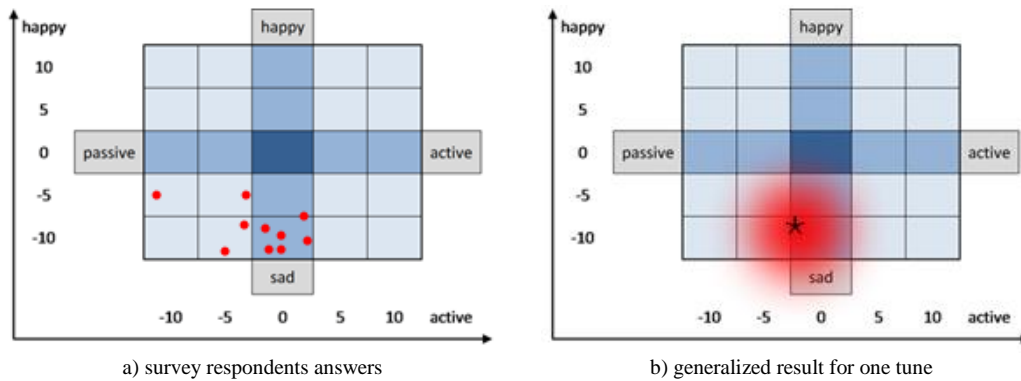


**Figure 2.** Created emotions plane for determining emotions of listening to a tune and calculate adjacent emotions.

In figure 3 is presented example of tune's questionnaire results. Red points represent emotions that feel respondents after listening to the tune in figure 3a. Figure 3b represents final result for performed tune we obtained as a mean value, which is presented as an asterisk with value (-1.8; -7.2). Variance which is represented as red

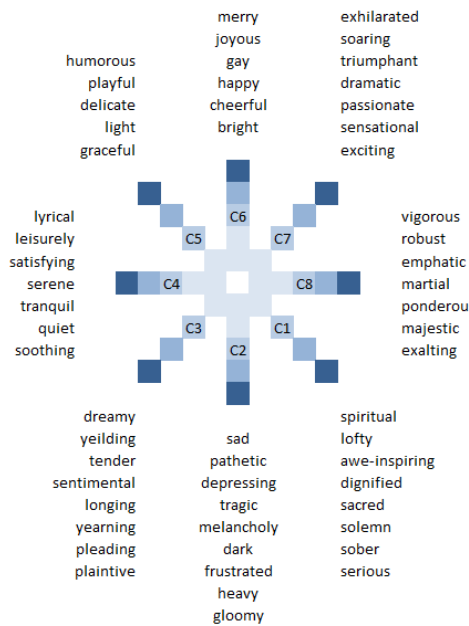


area in the active axis and corresponds to 3.48 value with a standard deviation equal to 1.86.



**Figure 3.** Questionnaire example result for tune popular music piece Amy Winehouse - Back to black.

For better understanding we also provided detailed description of plane areas as a supplementary material, which is based on Hevner’s adjective circle and shown in figure 4.

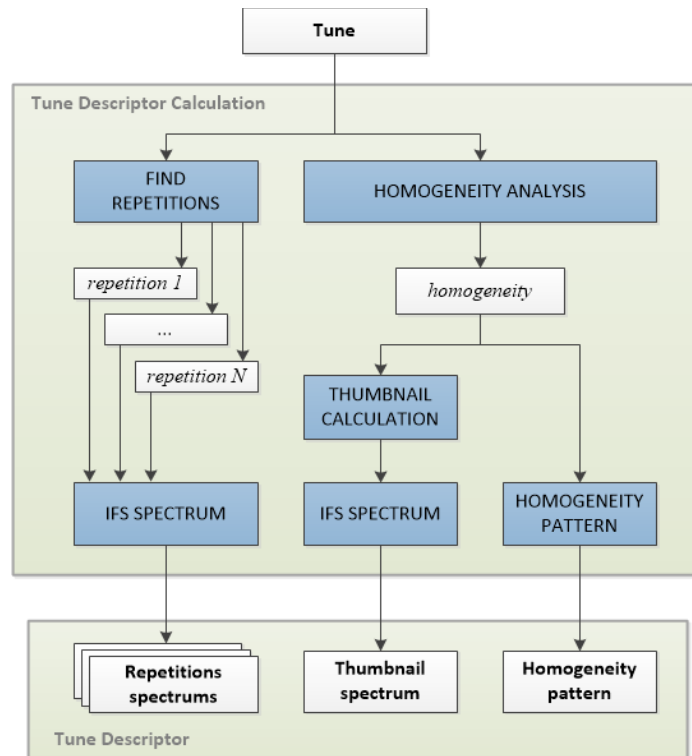


**Figure 4.** Detailed description of emotion’s plane based on Hevner’s adjective circle.

#### 4. Tune’s descriptor

The approach for tunes comparison is extremely important for system successful operation. We based the approach for tunes comparison on the method described in [2], which compares tunes by their descriptors that contain spectrums of most repeated parts or in other words about repetitions inside the tune.

Tune's descriptor calculation is outlined in figure 2, where white boxes show data and blue boxes show processing steps. Acoustic data is used as an input. Output is a tune's descriptor, which contains three significant parameters: tune's repetitions, thumbnail, and homogeneity pattern.



**Figure 5.** Tune's descriptor contains information about repetitions, thumbnail and tune's internal homogeneity.

The idea for acoustic signal comparison consists in using instantaneous frequency spectrum. That means that we compare IFS spectrums of signals when we want to compare one tune to another. In the initial work only repetitions were used, but we are proposing to extend the descriptors with tune internal homogeneity information and enrich the list of most representative parts.

Instantaneous frequency spectrum calculation, tune's repetitions detection, and tune internal homogeneity calculation approaches are described in chapter 4.1, 4.2 and 4.3 accordingly.

#### 4.1. Instantaneous Frequency Spectrum

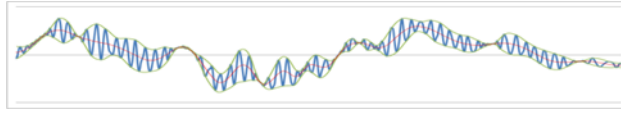
The HHT is a way to decompose a signal into so-called intrinsic mode functions (IMFs) using Empirical Mode Decomposition (EMD) [8] and then obtain instantaneous frequency data by means of the Hilbert transform (HT) [14]. Since the decomposition is based on the local characteristic time scale of the data, it can be applied to nonlinear and non-stationary processes.

The EMD method is a necessary step to reduce any given data into a collection of IMFs to which the Hilbert transform can be applied to.

To extract IMFs from the signal  $X(t)$ , all local extrema (minima and maxima) should be found first. Then we should create an upper envelope  $e_u(t)$  by local maxima and a lower envelope  $e_l(t)$  by local minima.

Envelopes are built by cube-spline interpolation. Using the upper and lower envelopes, the mean  $m(t)$  is calculated as

$$m(t) = \frac{e_u(t) + e_l(t)}{2} \quad (1)$$



**Figure 6.** Signal (blue), its envelopes (green) and mean (red) by envelopes

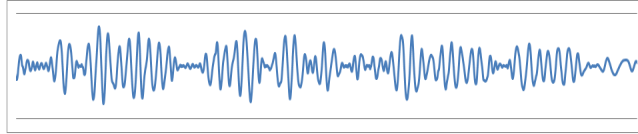
The result is shown in figure 6. The difference between the data and  $m(t)$  is the first component  $h_1(t)$ , which represents *proto IMF*. An IMF is defined as a function that satisfies two requirements:

- 1) the number of extrema and the number of zero-crossings must either be equal or differ at most by one,
- 2) at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

Until  $h_1(t)$  does not satisfy the definition of the IMF mentioned above, it should be iteratively refined using the same procedure. Thereby for  $h_1(t)$  we get next component  $h_2(t)$  and then  $h_3(t)$  and so on until stop criteria (2) becomes true, where  $\varepsilon$  is a small number. In this work  $\varepsilon$  was set to 0.0001.

$$\frac{\sum_t (h_k(t) - h_{k-1}(t))^2}{\sum_t (h_{k-1}(t))^2} < \varepsilon \quad (2)$$

After repeated refinement up to  $k$  times,  $h_k(t)$  becomes the first IMF of the signal, called  $c_1(t)$ . Figure 7 shows the first IMF obtained from the data in the figure 6.

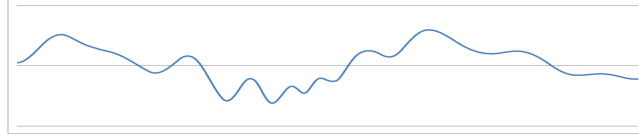


**Figure 7.** First IMF obtained from the signal

After we obtain the first IMF, we can get the residue  $r(t)$  by subtracting  $c_1(t)$  from initial data:

$$r(t) = X(t) - c_1(t) \quad (3)$$

The residue of initial signal from the figure 6 is shown in the figure 8.



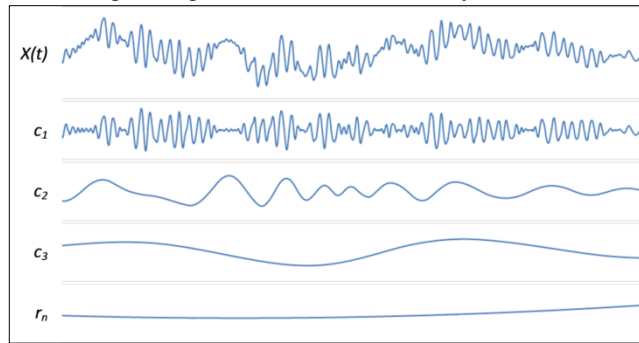
**Figure 8.** Residue after subtracting first IMF  $c_1$

In the next round of the sifting process the residue  $r(t)$  is considered as a signal  $X(t)$  and the sifting procedure is repeated the same way to obtain  $c_2(t)$ , then  $c_3(t)$ , and so on until residue becomes a monotonic function without extrema. When we sum all obtained IMFs with the last residue, we get initial data signal as follows:

$$X(t) = \sum_{i=1}^n c_i + r_n \quad (4)$$

The good feature of such decomposition is that each IMF represents an intrinsic component of the real physical effect.

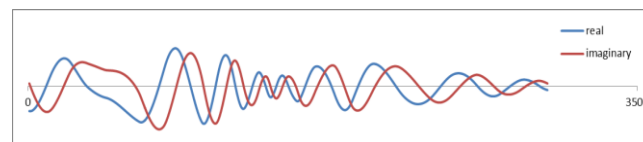
Figure 9 shows the original signal and IMFs obtained by means of EMD.



**Figure 9.** The resulting empirical mode decomposition components from the music data: the original data  $X(t)$  and the components  $c_1 - c_3$ ;  $r_n$  is a trend

### *Hilbert Transform*

The Hilbert transform can be interpreted as a phase shifter, which changes the phase of all frequency components of a signal to  $\pi/2$ . To shift a phase, the initial signal is processed with a Fourier transform and then every component of the resultant spectrum is multiplied by imaginary  $i$  and the spectrum is converted back to signal using the inverse Fourier transform. An example of the original signal and derived signal with shifted phase are shown in figure 10.



**Figure 10.** Initial Signal (blue) and obtained imaginary signal after HT (red)

The imaginary signal  $\tilde{X}(t)$  is orthogonal to original signal  $X(t)$ . This feature allows us to develop from  $\tilde{X}(t)$  and  $X(t)$  a complex analytical signal  $H(t)$ :

$$H(t) = X(t) + i\tilde{X}(t) \quad (5)$$

$H(t)$  is described as a vector on the complex plane where  $X(t)$  and  $\tilde{X}(t)$  are projections to real and imaginary axes, respectively.

The advantage of this representation is that we have an opportunity to determine instantaneous parameters of the signal  $H(t)$ , i.e., the amplitude and frequency, where the radius of each circle represents the amplitude and the space between circles means the frequency.

Instantaneous amplitude is calculated as complex number length:

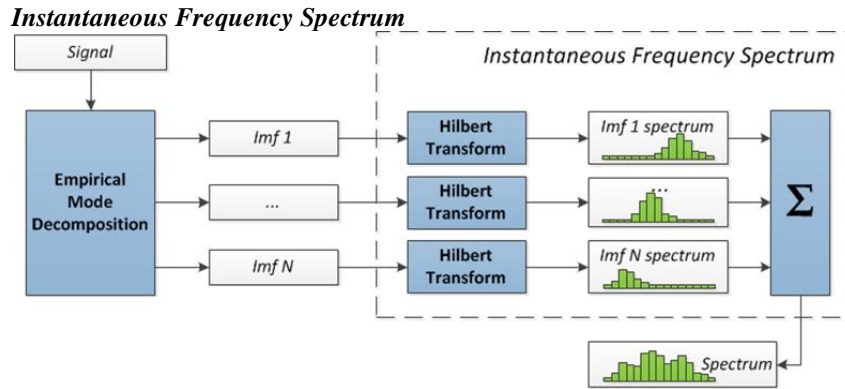
$$A(t) = \sqrt{(\text{real}H(t))^2 + (\text{imag}H(t))^2} \quad (6)$$

Instantaneous frequency is calculated as instantaneous phase derivative of a signal:

$$f(t) = \frac{1}{2\pi} \varphi'(t), \quad (7)$$

where phase  $\varphi$  is calculated as

$$\varphi(t) = \tan^{-1} \frac{\text{real}H(t)}{\text{imag}H(t)} \quad (8)$$



**Figure 11.** Scheme for calculating the IFS.

The IFS calculation method is outlined in the figure 11. White boxes show data and blue boxes show processing steps. As inputs, we use a number of IMFs that represent intrinsic functions of the same signal. As an output, we get a histogram of amplitudes by frequencies. A histogram for every IMF is calculated in the same way. For each IMF, we get instantaneous frequencies and instantaneous amplitudes using the Hilbert transform as described before. These frequencies and amplitudes are used to create a histogram. Formally, this is described as

$$b_i = \sum_t A(t), \quad (9.1)$$

$$t: \beta(i-1) \leq \log f(t) \leq \beta(i),$$

$$i = \overline{1, N}$$

$$\beta(i) \stackrel{\text{def}}{=} \frac{i}{N} \log F_{max} \quad (9.2)$$

where

- $b_i$  is height of  $i$ -th bar of the histogram,
- $A(t)$  is an instantaneous amplitude at time  $t$ ,
- $f(t)$  is an instantaneous frequency at time  $t$ ,
- $\beta(i)$  is a frequency upper boundary for  $i$ -th histogram bar,
- $N$  is a number of bars in the histogram,
- $F_{max}$  is maximal frequency.

#### 4.2. Repetitions

The more repetitions and similar phases there are in a piece of music, the easier it is for people to have affinity for it. Since repeated parts are very important, we identify repetitive structures in a tune by using a self-similarity matrix. We found most outstanding repetitions of a tune to calculate their IFS spectrums for including this information into a tune descriptor. The approach of finding repetitive structures within music pieces was initially introduced in [9].

##### ***Finding Repetitions***

The tune is divided into frames of size 0.7 seconds and with a shift size 0.1 second. Each of these frames is converted using MFCC and used to calculate distance to each other. A distance equal to 0 means that the fragments are identical; the greater the value is, the less similar the fragments are. From these values we get a self-similarity matrix showing the results of pair-wise comparisons of all frames. This matrix can be represented in the form of a grayscale image where each number is represented by brightness. Black points in the image shows the value of 0. Further processing is based on the data from this matrix.

In the self-similarity matrix, similar parts in a tune appear in the form of dark lines. We take a small window and consider only the values within it. The distances at the points marked with letter A should be significantly smaller than those at the points marked with letter B. If this is true, considered area would look like a line, so points marked with letter A are a part of a repetition, see figure 12. To avoid division by zero, a very small constant  $\varepsilon$  is added to the numerator and denominator (about  $1 \times 10^{-5}$ ). Thus, the lower the value we get using this equation, the more desired the line is.

$$f = \frac{A_1 + 2 * A_2 + A_3 + \varepsilon}{B_1 + B_2 + B_3 + B_4 + \varepsilon} \quad (10)$$

	B <sub>4</sub>	A <sub>3</sub>
B <sub>2</sub>	A <sub>2</sub>	B <sub>3</sub>
A <sub>1</sub>	B <sub>1</sub>	

Figure 12. Window size 3x3

According to (10), if the values at points marked with the letter A are less than the values at points B, the result is less than 1. Thus, if the result is in the range from 0 to 1, it is a potential line. Values closer to 0 indicate a stronger line, and values closer to 1 show a weakly visible line. We are not interested in values greater than 1, so they can simply be replaced by 1, where 1 means the situation when the sum of A-points is approximately equal to the sum of B-points, meaning that considered area points inside the window does not look like a line. After this process, places with lines are strongly emphasized and it is easy to see them. An example is shown on figure 13 b.

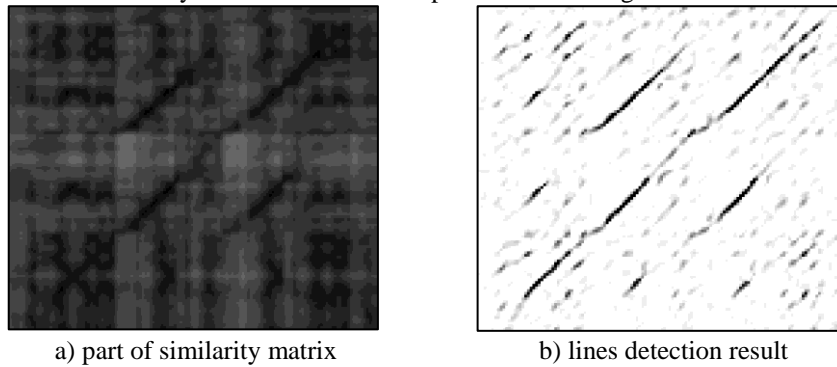


Figure 13. Line parts detection results.

After line detection with a 3 by 3 window, we can better see the lines we are about to detect. These lines vary in brightness (See figure 14). Darker lines indicate a strong similarity between corresponding musical fragments. Lighter gray lines indicate less accurate repetition, which may be because of differences in tone, in performance, or because of different musical instruments.

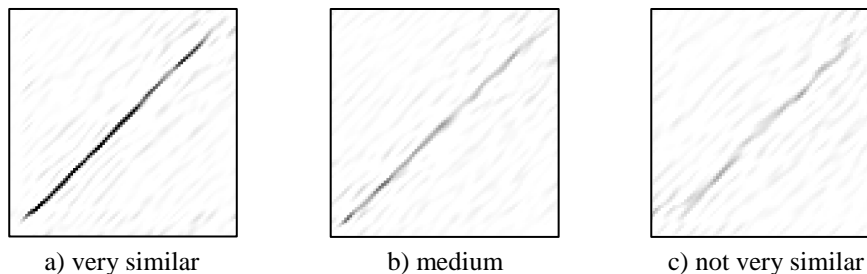
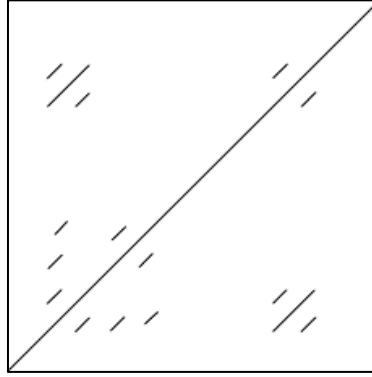


Figure 14. Different similarity level comparison.

When there is at least a light line of repetition that can be seen in an image, the listener should be able notice similarity between these fragments.

Finally, to detect lines we iterate through all closed sets of dark points and search for coordinates of the bottom left point and the top right point within every set. These are the boundaries of lines – the result of the line search in the image. When a line with coordinates  $(x_1, y_1) - (x_2, y_2)$  is found, this means that the fragment  $(x_1 - x_2)$  is similar to the fragment  $(y_1 - y_2)$  within this piece of music. After selecting the most significant lines from all found lines, we get a picture such as shown in figure 15.



**Figure 15.** Example of most significant repetitions found.

#### 4.3. Tune Internal Homogeneity Detection

Tune homogeneity is information about internal structure of a tune. Homogeneity for any specific moment within a tune is considered as high when this moment sounds typical to the whole tune. The approach of determining tune's internal homogeneity and thumbnail was initially introduced in [5]. This information has time dimension, so that we can see how high the homogeneity is for any moment of the music piece. For example, a music piece is played by piano slowly from the beginning till the end, but there are a few short inclusions of irregular musical instrument performances such as drums parts. In that case most of the time tune is typical to itself and can be called homogeneous, but fragments with drums sound untypical and are called inhomogeneous.

The achievements from [5] were used in this work to find outstanding and important parts within processed tunes. Information about the most homogeneous parts within tunes are used to build the descriptor. The most homogeneous part describes the most typical fragment of a tune that we called as a thumbnail.

##### **Homogeneity calculation**

Tune homogeneity detection can be described as following steps.

Step 1. Split tune into  $N$  frames  $a_i$  of  $T$  seconds size:

$$a_i, i = \overline{1, N}.$$

Step 2. Calculate IFS spectrum  $s_i$  for every frame  $a_i$ :  $s_i = \text{IFS}(a_i), i = \overline{1, N}$ .

Step 3. Calculate average spectrum  $s_a$  of all spectrums  $s_i$  in (11).

$$s_a(k) = \frac{1}{N} \sum_{i=1}^N s_i(k), \quad k = \overline{1, M} \quad (11)$$



where  $k$  is a spectrum bar number,  $M$  is a spectrum bars count.

Step 4. Calculate differences  $D = \{ d_i \}$  for every frame spectrum  $s_i$  with overall spectrum  $s_a$  according to (12).

$$D \stackrel{\text{def}}{=} \{d_i = \text{dist}(s_i, s_a), \quad i = \overline{1, N}\} \quad (12)$$

Spectrums difference in step 4 is calculated as follows as shown in (13), where  $x$ ,  $y$  are IFS spectrums.  $x(k)$ ,  $y(k)$  are  $k$ -th frequency range value in IFS spectrum,  $M$  is number of frequency ranges in  $x$  and  $y$ .

$$\text{dist}(x, y) = \sum_{k=1}^M (x(k) - y(k))^2 \quad (13)$$

After described 4 steps we get differences  $D = \{ d_i \}$ .  $D$  is a first approximation of homogeneity result and it can be further refined.

#### **Iterative refinement**

First homogeneity approximation  $D$  is calculated from average spectrum  $s_a$  which contains information not only from homogenous parts, but also from inhomogeneous parts. Now when we already know which frames are inhomogeneous, we can exclude them from average spectrum  $s_a$  to make the result more accurate.

Step 5. Set weight  $w_i$  for every frame  $a_i$  depending on their homogeneity so that higher distance  $d_i$  leads to lower weight  $w_i$  as in (14).

$$w_i = \frac{\max_{j=\overline{1, N}} d_j - d_i}{\max_{j=\overline{1, N}} d_j - \min_{j=\overline{1, N}} d_j}, \quad i = \overline{1, N} \quad (14)$$

Step 6. Calculate weighted average spectrum  $s_{wa}$  of all spectrums  $s_i$  according to (15).

$$s_{wa}(k) = \frac{\sum_{i=1}^N w_i * s_i(k)}{\sum_{i=1}^N w_i}, \quad k = \overline{1, M} \quad (15)$$

Step 7. Calculate differences  $D = \{ d_i \}$  for every frame spectrum  $s_i$  with overall weighted average spectrum  $s_{wa}$  as in (16).

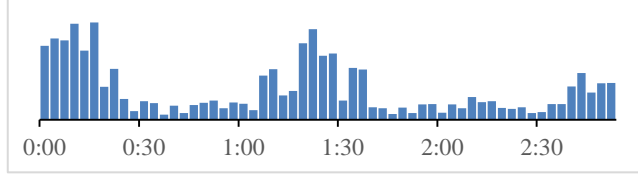
$$D \stackrel{\text{def}}{=} \{d_i = \text{dist}(s_i, s_{wa}), \quad i = \overline{1, N}\} \quad (16)$$

Step 8. Repeat steps 5, 6, 7 until differences  $d_i$  stop changing. Stop criteria can be defined as in (17), where  $r$  – iteration number,  $d_i^{(r)}$  – difference  $d_i$  on iteration  $r$ .

$$\frac{\sum_{i=1}^N (d_i^{(r)} - d_i^{(r-1)})^2}{\sum_{i=1}^N (d_i^{(r-1)})} < \varepsilon \quad (17)$$

In step 8,  $\varepsilon$  is a minimal relative change size that can be set to 0.01 meaning 1% change.

*Visual presentation.* Homogeneity analysis result  $D$  can be visualized as a bar chart where horizontal axis represents time, and bars represent homogeneity  $d_i$  of the corresponding fragment. An example is shown on figure 16.



**Figure 16.** Homogeneity result example.

The higher is a bar the greater difference to the tune it represents. On the other hand, shortest bars show most homogeneous parts of the tune.

### ***Tune thumbnail***

Tune thumbnail is the most representative part of a tune and it is essential, because by listening to a thumbnail of the tune people can easily recall that tune from their memory and fillings they were experienced during it performance. It hence how our memory is working to memorize some significant parts from the music pieces. So we use tunes thumbnails as a second parameter in the tunes descriptors.

Let's say that we need a thumbnail of length  $T_{th} = 30$  seconds. To get most representative 30-seconds part, we use a sliding window on previously calculated differences  $D = \{ d_i \}$ . For the case when window size  $T_{th} = 30$  seconds, and frame size  $T = 3$  seconds, sliding window contains  $L = T_{th} / T = 10$  items of  $d_i$ . Shifting window with step  $T$  we calculate sum of items  $d_i$  within the window.

Tune fragment, which correspond to the minimal sum value is the most representative fragment for tune thumbnail (18).

$$Thmb_{start} = \arg \min_i \sum_{j=i}^{i+L} d_j, \quad (18)$$

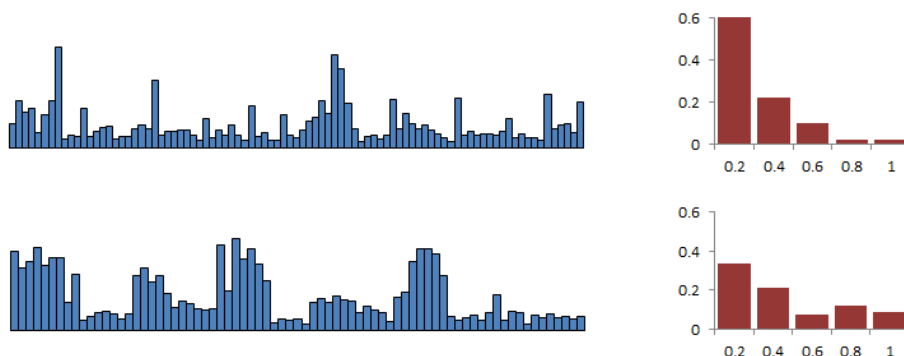
$$i = \overline{1, N - L}$$

where  $L$  is a sliding window size,  $Thmb_{start}$  is a beginning of the thumbnail.

### **4.4. Tune Homogeneity Pattern**

Tune homogeneity pattern is the information about internal structure of a tune, which describes how often tune changes the way of its performance. For example, when internal homogeneity of the music piece changes frequently, listeners may memorize it as highly varying tune, so the emotions they will experience will be exiting and interesting compare to the tunes, which are performed in monotonous way. To make it comparable between tunes we decided to present it as a histogram that shows the frequency of different homogeneity values within tune internal homogeneity result.

In this work we set the homogeneity pattern histogram to consist of 5 bars. Figure shows two examples of homogeneity pattern for two different tunes.



**Figure 17.** Two examples of tunes internal homogeneity (blue) and their homogeneity patterns (red).

For the tune above in figure 17, we can see that most homogeneity bars (blue) are short. That fact can be seen on corresponding homogeneity pattern histogram (red): the probability of low values is very high. The tune below in figure 17 has less homogeneous parts compared to the previous tune. Its homogeneity pattern looks a bit different and has higher probability of greater values (red).

## 5. System working steps

The operation of the system assumes that the database has enough records, containing various tunes from all considered areas: western popular music and classical genres for the current paper. It is also assumed that every tune in the database has true values of the tune's emotion. In this work we tried to accomplish this by using estimates from multiple experts. The detailed description how a test tune emotion is determined within the system is described below in this chapter.

### 5.1. Tune emotional effect estimation

Tune processing within the system has two main parts: finding matching tunes in the database and corresponding emotions combining. In the first part we search for multiple matches instead of a one single best match for better result precision since more related records are aggregated to calculate the result. Such approach is helpful for reducing the effect of outstanding values. In the second part we wisely combine emotions from all matches to get the result emotion for the test tune.

#### 5.1.1. Tunes comparison

Tunes matching within the database is a challenging task by itself. The approach we used in this paper is based on the one described in [2]. Every tune is processed to get the descriptor which is like a fingerprint that precisely identifies the characteristics of the tune but at the same time is small and easy to process, compare and store. As described before in this work tune descriptor contains a set of fields: repetitions spectrums, thumbnail spectrum and a homogeneity pattern. Every field is compared separately and has its own weight. Table 2 describes descriptor fields, comparing approaches and weights.

Table 2. Descriptor fields comparison approaches

	Field	Comparison approach	Weight
1.	Repetitions spectrums	AVG <sub>k</sub> for every to every repetition spectrum comparison as described in [2]	40%
2.	Thumbnail spectrum	Euclidian distance	30%
3.	Homogeneity pattern	Euclidian distance	30%

Comparison result is a tunes difference that is calculated as a sum of weighted differences from every field comparison as shown in (19).

$$d_j = \sum_i w_i * diff(TT.field_i, T_j.field_i) \quad (19)$$

where  $d_j$  is a difference between test tune  $TT$  and  $j$ -th tune from the database,

$TT$  is a test tune,

$T_j$  is a  $j$ -th tune from the database

$TT.field_i$  is a  $i$ -th field in the  $TT$  tune's descriptors field,

$w_i$  is a weight of  $i$ -th field.

### 5.1.2. Matching tunes selection and emotions combining

When a new test tune is provided to the database to find matching records, its descriptor is compared to all records in the database.  $N$  most matching tunes are selected, i.e.  $N$  tunes with the smallest difference values compared to the test tune. In this work we set  $N = 10$ .

Matching tunes emotions are combined with respect to the match distance in the way where tunes with smaller distance have higher weight as:

$$\gamma^* = \frac{\sum_{i=1}^N w_i * \gamma_i}{\sum_{i=1}^N w_i} \quad (19.1)$$

$$w_j = \left( \frac{\max_i d_i - d_j}{\max_i d_i - \min_i d_i} \right)^k \quad (19.2)$$

where  $\gamma^*$  is a combined emotion,

$\gamma_i$  is an  $i$ -th record's emotion within  $N$  most matching tunes,

$w_i$  is an  $i$ -th record's weight within  $N$  most matching tunes,

$d_j$  is a  $j$ -th record's distance to the test tune.

### 5.2. Database extension and results refinement

Every time tune's emotional effect is estimated, tune can be added to the database if the user can provide correction to the auto-calculated emotional effect. Such addition of a new tune is an important correction to the system that refines the following results provided. The more tunes the database contains, the better correct percentage the system shows.

In this work we were extending the database with new tunes that were evaluated by multiple experts. For personal use of the system the only single user can extend the database with every new tune by simply setting his/her emotion or by correcting the auto-calculated one while the person is listening to the tune.

## 6. Results analysis

In this chapter we describe the results we got while validating the system. At first, the database was filled with data of 20 music pieces and validated using 10 other tunes. Later we expanded the database several times and studied how the system efficiency grows. Experiments and results are described below.

### 6.1. Experiments parameters

For system efficiency evaluation 10 different tunes were taken. Every tune from evaluation list was provided for the system to get the calculated emotional effect. That calculated emotional effect is then compared to existing emotional effect that was achieved during the survey.

### 6.2. Initial percent of correct selection

Table 3 shows the comparison results for 10 test tunes from evaluation list, where the emotions comparison was done as Euclidian distance on emotions plane described above in chapter 3.1. Distance equal to 0 is taken as 100% accuracy, distance equal to 20 or greater is taken as 0% accuracy (since emotions plane's width and height is taken equal to 20).

**Table 3.** Test tunes evaluation results

Tune number	Emotional distance	Accuracy
Tune 1	6.48	69%
Tune 2	7.64	62%
Tune 3	6.72	68%
Tune 4	8.44	57%
Tune 5	9.60	53%
Tune 6	5.44	74%
Tune 7	10.72	48%
Tune 8	8.20	60%
Tune 9	10.92	47%
Tune 10	7.80	62%
<b>Average</b>	7.40	61%

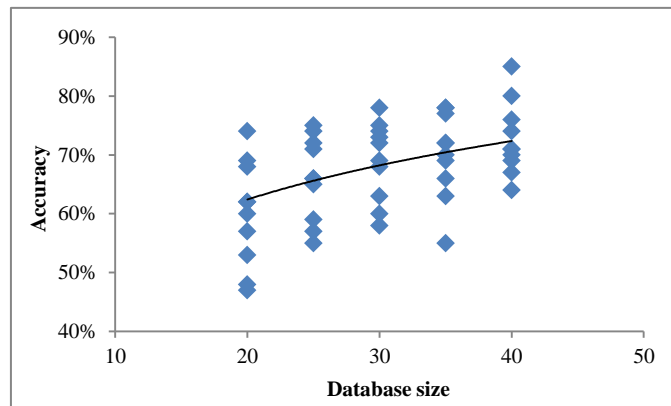
The overall efficiency was calculated as average and is equal to 61%. As we show later in this paper, the value goes higher as we increase the database size.

### 6.1. Correct percentage growth trend on database extension

We were iteratively increasing the database size to study the efficiency growth of the system. The results show that the efficiency of the system depends on the database size. The table 4 shows the results for 5 experiments with different database sizes. Figure 18 visualizes the dependency visually.

**Table 4.** Efficiency growth of the system

Database size	20	25	30	35	40
<b>Tune 1</b>	69%	66%	69%	72%	71%
<b>Tune 2</b>	62%	57%	58%	55%	70%
<b>Tune 3</b>	68%	74%	78%	70%	80%
<b>Tune 4</b>	57%	72%	74%	69%	67%
<b>Tune 5</b>	53%	65%	73%	77%	71%
<b>Tune 6</b>	74%	55%	60%	66%	64%
<b>Tune 7</b>	48%	75%	63%	72%	74%
<b>Tune 8</b>	60%	59%	68%	78%	69%
<b>Tune 9</b>	47%	71%	75%	78%	85%
<b>Tune 10</b>	62%	66%	72%	63%	76%
<b>Average</b>	<b>61%</b>	<b>66%</b>	<b>69%</b>	<b>71%</b>	<b>72%</b>



**Figure 18.** Positive trend of system's efficiency depending on database size.

The trend of correct results on increasing database size shows a logarithmic curve.

The achieved result can be considered as very good. The database with only 40 tunes gave 72% accuracy. That is equal to 3.75 distance on emotions plane. On Hevner's emotions categories this would mean that in most cases calculated results are either equal to the surveyed values, or differ to one neighboring category that is similar. The results must be even better if more people are used in the surveying, and bigger amount of tunes are included in database. Since it is hard to achieve in a simple survey, we described the possible way of database extension and results correction on the fly when system is already in use. For the paper we have shown that the approach gives

appropriate results even for small amounts of data. System parameters optimization to reach the highest possible accuracy is planned in the future work.

## 7. Conclusion and Future Work

In this paper we proposed the musical tunes emotions identification system by means of intrinsic musical characteristics that is based on experts' evaluations in a survey. We define three significant tunes parameters: repetitions inside a tune, thumbnail of a music piece, and homogeneity pattern of a tune, because they are related to how people perceive music pieces and we can express the essential features of emotional aspects of each piece.

Our system consists of music-tune features database and computational mechanism for comparison between different tunes. Based on Hevner's emotions adjectives groups we created a way of emotion presentation on emotion's plane with two axes: activity and happiness. That makes it possible to determine perceived emotions of listening to a tune and calculate adjacent emotions on a plane. The approach for estimating a tune emotion effect uses the database and aggregates the emotion values among tunes similar to the processed one.

We performed a set of experiments on western classical and popular music pieces, which presented that our proposed approach reached 72% precision ratio and show a positive trend of system's efficiency when database size is increasing. The results show that even a database of 40 tunes gives a very good result.

The system can have a big variety of applications from personal use to global social environment maintenance. In personal use the system can create a playlist according to the listener's mood for the specific moment. In more global applications we can suggest to use the approach to influence on people in public places such as shops, airports and stations. Besides that the system can be extended with multiple emotions databases specific to different cultures. That will help in studying the cultural differences which in turn will lead to better cross-cultural understanding.

In future work we are planning to optimize system parameters to reach the highest possible accuracy. Within that we are planning to improve IFS spectrums comparison by taking into account human ear specifics. As another direction we are planning to modify and extend the system for researching on how visual data such as images and video may affect the emotions when shown alongside with the music.

## References

- [1] K. Hevner, Experimental Studies of the Elements of Expression in Music. *American Journal of Psychology*, Volume 48, pp. 246-268, 1936.
- [2] T.Endrjukaite and Y. Kiyoki, Music Similarity Analysis through Repetitions and Instantaneous Frequency Spectrum. Proceedings of the 5th International Conference of Signal Processing Systems Vol. 1, No. 2, December 2013.
- [3] J. Foote, Visualizing music and audio using self-similarity, Proceedings of the 7th ACM International Conference on Multimedia (Part 1), pp. 77-80, October 30-November 05, 1999.
- [4] B. Logan, Mel Frequency Cepstral Coefficients for Music Modeling. In Proceedings of the 1<sup>st</sup> International Symposium on Music Information Retrieval, 2000.
- [5] T.Endrjukaite and Y. Kiyoki, Music Homogeneity Analysis through Instantaneous Frequencies, In Proceedings of the 11th International Conference on Advances in Mobile Computing & Multimedia, 2-4 December, 2013.

- [6] J. Sloboda, *Exploring the musical mind: cognition, emotion, ability, function*, Oxford University Press, a book in English, 2005.
- [7] T. Taniguchi. *Ongaku to Kanjoh (Music and Emotion)*. Kitaohji-Shobo, a book in Japanese, 1998.
- [8] N. Huang et al. *The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis*. Article in Proc. R. Soc. Lond. A, 8 March, 1998, vol. 454 no. 1971, pages 903-995.
- [9] T. Endrjukaite and N. Kosugi, Music visualization technique of repetitive structure representation to support intuitive estimation of music affinity and lightness, *Journal of Mobile Multimedia, Volume 8 Issue 1, (2012) 049-071*.
- [10] D. Levitin. *This Is Your Brain on Music: The Science of a Human Obsession*. Dutton Adult, a book in English, 2006.
- [11] T. Endrjukaite and N. Kosugi, Time-Dependent Genre Recognition by means of Instantaneous Frequency Spectrum, *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*, 2012.
- [12] G. Tzanetskis and P. Cook. 2002. Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, 10(5), 2002.
- [13] P. Lindsay, N. Donald. *Human Information Processing: Introduction to Psychology*. Academic Press Inc; 2nd edition, a book in English, 1977.
- [14] N. Huang and S. S P Shen. *Hilbert–Huang Transform and Its Applications. Interdisciplinary Mathematical Sciences; volume 5*. World Scientific Publishing Co. Pte. Ltd.; a book in English, 2005.



# Human Reaction in Thailand based on Social Media Analysis after the East Japan Great Earthquake

Takako Hashimoto<sup>a</sup>, Teeranoot Chauksuvanit<sup>b</sup>, Supavadee Aramvith<sup>b</sup>  
and Yukari Shiota<sup>c</sup>

<sup>a</sup>*Chiba University of Commerce, Chiba, Japan*

<sup>b</sup>*Chulalongkorn University, Bangkok, Thailand*

<sup>c</sup>*Gakushuin University, Tokyo, Japan*

**Abstract.** After the East Japan Great Earthquake occurred in Japan on 11th March 2011, a large number of messages related to the earthquake were posted to Thai social media web sites. There are a lot of different topics concerning about the earthquake that were recognized as Thai people's reactions to the earthquake. Therefore, exploring topics and messages related to the earthquake on Thai's social media gains rich insights into the Thai social contexts. The goal of this research is to analyze Thai people reactions to the East Japan Great Earthquake on Thai social media. This paper explores messages from 3 well-known web sites in Thailand, and analyzes how people reacted to the earthquake related Thai society and culture.

**Keywords.** Thailand and Japan, Human Reaction, Social Media, East Japan Great Earthquake

## Introduction

Social media in which individual users post their opinions and gradually build their consensus, is recognized as one of the important collaborations in today's information oriented society. After the topical problems like a disaster, people's behavior is influenced by this collaboration. Especially after the East Japan Great Earthquake occurred in Japan on 11th March 2011, a large number of messages related to the earthquake were posted to social media such as Twitter, Facebook, and so on, not only in Japan but also all over the world. Particularly, in Asian countries, a lot of topics concerned about the earthquake were observed. Exploring topics related to the earthquake on social media in Asian countries gains a rich insight into the Asian social context after the earthquake. The goal of our research is to analyze Asian people reactions to the East Japan Great Earthquake on social media using data mining technique.

We already proposed the graph based topic extraction method[1,2]. In our method, first, social media messages are crawled and keywords are extracted using morphological technique. Next, we construct a snapshot document-term matrix at each time stamp. Then, we investigate topic transitions over time by forming network graphs

from the matrix. Our method could show the time series structure transition by network graphs, so that we could extract topics and their changes over time.

Our method is based on keywords. Once keywords were extracted, the method can be language-independent. In this paper, we tried to apply the method to other languages. As the first target, we selected Thai language. In Thailand, social media is quite popular. After the East Japan Great Earthquake, a lot of people posted their messages related to the earthquake to social media. We already tried the preliminary approximation for crawling messages and extract keywords from one Thai social media site[3]. In this paper, as the next step, we analyze topics that show how Thai people reacted to the earthquake by forming topic structures and compare with Japan reactions.

This paper is organized as follows. Section II introduces our already-proposed method to explore topic structures on social media from network graphs. Section III presents the situation of social media in Thailand and selects target social media for this work. In Section IV, our method is applied to Thai social media. Section V compares reactions on social media between Thailand and Japan. Section VI refers to existing researches. Finally, Section V concludes this paper.

## 1. Our Topic Extraction Method

Our method that was already proposed previously consists of the following 4 steps [1, 2, 3, 8]:

- STEP A: Crawling social media messages and extracting significant terms.
- STEP B: Constructing co-occurrence networks.
- STEP C: Clustering and organizing topic structures.
- STEP D: Tracing topics over time.

Each step is described below with some examples derived from a blog titled “Banya Nippou” about affected people’s needs provided by the non-profit organization[4].

### 1.1. STEPA: Crawling social media messages and extracting significant terms

STEP A crawls messages  $D = \{d_i\}$  from social media. One message is defined as one document and the step retrieves it as the following tuples:

$$d_i = (MID_i, Posted_i, Title_i, Content_i)$$

Here,  $MID_i$  is an ID of each document,  $Posted_i$  is a date-time that each document was posted,  $Title_i$  is a title of each document and  $Content_i$  is a text of each document.

The step then extracts terms that are nouns, verbs, adjectives, and adverbs from  $Content_i$  of each  $d_i$  by morphological analysis, and the score of an individual term in  $d_i$  is calculated using RIDF (residual IDF) [4] measure. Finally, the words with high RIDF value are selected as a list of keywords  $KW = \{kw_{ij}\}$ . Table I and Table II show some examples of crawled messages and extracted keywords.

TABLE I. EXAMPLE OF CRAWLED MESSAGES BY STEP A

<i>MID</i>	<i>Posted</i>	<i>Title</i>	<i>Content</i>
1	2011/06/03	Problem	* No delivery of supplies to place that is closed to cars. * No supply deliveries for places that are closed to cars. * Shops are too far. * Want to live in original places.....
2	2011/6/4	Problem	* Very cold due to lack of stoves at the evacuation center. * Feel fear about the future (2-3 years later). * No money for buying clothes. * Can't go shopping without cars.....
3	2011/6/5	Problem	* No information at the evacuation center. * No enough foods. No space for beds. * Can't have maternity goods. * No enough toilets. * No clothes. No snacks for kids. * Every day, we have cup noodles only.....
...	...	...	...
15	2011/6/30	Request	* Want to move to temporary house. * Feel fear for the future.* Need clothes for summer. * Want to find jobs that even old people can do.....
...	...	...	...

TABLE II. EXAMPLE OF KEYWORD EXTRACTION BY STEP A

<i>MID</i>	<i>Posted</i>	<i>Keywords (I shows R IDF value)</i>
1	2011/06/03	Car [0.13], Place[0.11], Supply[0.38], Delivery[0.13], No[0.83], Shop[0.12], Far[0.14]....
2	2011/6/4	Delivery[0.12], No[0.33], Fear[0.16], Evacuation[0.61], Center[0.13], Cold[0.12]....
3	2011/6/5	Information[0.13], Evacuation[0.41], Center[0.23], Food[0.11], Goods[0.11], Kid[0.24]....
...	...	...
15	2011/6/30	Move[0.1], Fear[0.26], Want[0.27], Job[0.29], Temporary[0.29], House[0.25]....
...	...	...

### 1.2. STEP B: Constructing co-occurrence networks

In STEP B, for  $\{kw_i\}$  outputted by STEP A, the posted date is delimited by an appropriate period (e.g. monthly, weekly, or daily), and  $D$  is grouped by the period as  $\{X_k\}$  and become time series data. STEP B then constructs a co-occurrence network  $G = \{g_k\}$  from  $\{X_k\}$ , so that the graphs are made in time-series.

Then, STEP B makes relevance network graphs of words appearing in each time series group of  $\{kw_i\}$ . Network graphs of related words are obtained using co-occurrence frequencies in a document, which is, for a subset  $X$  of  $D$  and terms  $w_{ij}$  and  $w_{ik}$ , if  $X$  contains  $w_{ij}$  and  $w_{ik}$ , then  $w_{ij}$  and  $w_{ik}$  are connected by an edge. Fig. 1 shows an example of graph structure constructed by STEP B.

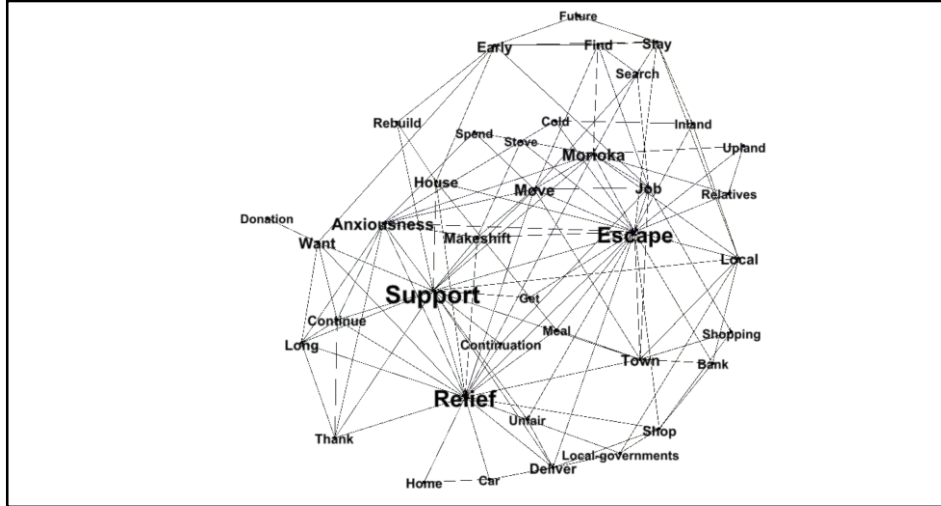


Fig. 1 An example of co-occurrence network

1.3. STEP C: Clustering and organizing topic structures

In our work, we define a community as a topic. In STEP C, we organize the topic structures for each snapshot network by using clustering techniques such as modularity measure[5] and LSA (Latent Semantic Analysis)[6] for forming communities[1, 8].

1.4. STEP D: Tracing topics over time

In STEP D, we computed topic similarities over time. As the similarity parameter, we adopted different similarity parameters such as Matthews correlation coefficient (MCC) [8] and cosine similarity. MCC is a measure of the quality for two binary classifications. Fig. 2 shows some examples of topic transition detection. We recognized the following topic transitions in social media.

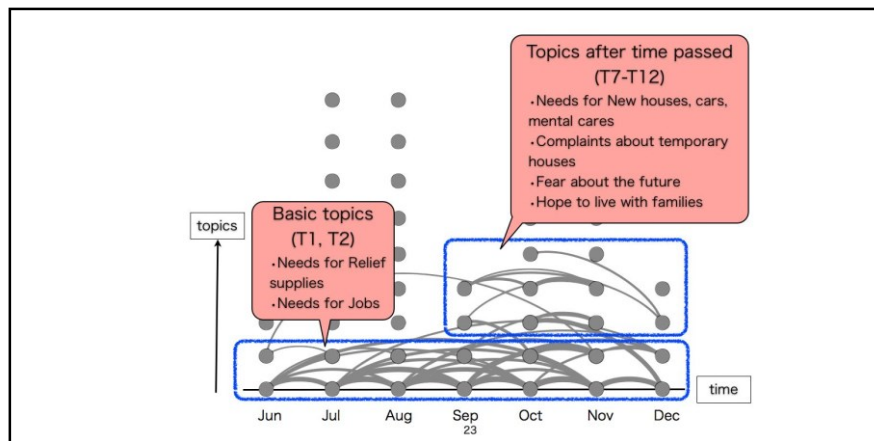


Fig. 2 An example of topic transition result

- B1 (Request for supply of goods, and Need of Job) is basic topics (needs) for afflicted people. They appear for long periods of time. In Fig. 2, the basic topic transitions express these basic needs.
- As time passed, people's needs gradually changed like "Need of new houses", "Complaints about temporary houses", "Needs of cars", "Needs of mental care", "Feel fear about the future" and "Hope to live with families". In Fig. 2, B2 shows the second main stream (topic transitions after time passed). This means that after things settled down, people wanted to rebuild their job, house, and life.

We believe our method helps to grasp topic change.

## 2. Social Media in Thailand

For applying our method to Thai language, first, we have to investigate the situation of Thai social media and select appropriate target social media. About social media in Thailand from ABAC pole on 24th of February 2012 [9] reported that Thai people use social media more than 90% and more than 46.9% use more than once in a day. The most popular social media in Thailand is [www.facebook.com](http://www.facebook.com) which shows rate 98.6 % for talking, update friends and news at the rate of 94.5%. The Internet Innovation Research Center listed 30 popular web sites got awards in Truehits.net Web Award 2011.

We examined 30 websites of Truehits.net Web Award 2011. Among those, we selected the following 6 websites as social media candidates in Thai.

- A) [www.kapook.com](http://www.kapook.com)
- B) [www.sanook.com](http://www.sanook.com)
- C) [www.pantip.com](http://www.pantip.com)
- D) [www.dek-d.com](http://www.dek-d.com)
- E) [www.facebook.com](http://www.facebook.com)
- F) [www.twitter.com](http://www.twitter.com)

According to our observation, we found that some websites can retrieve messages on the Great East Japan Earthquake but some cannot find any messages because of different purposes and structure of the sites. To crawl messages from Thai social media, Thai language keywords related to the East Japan Great Earthquake were set. We got start from 17 keywords to 96 keywords and found that some keywords were not effective to some social media candidates up. Keywords were adjusted many times in different words, form and sequence. Finally 21 Thai words were selected as appropriate keywords to crawl data (Table II), and 3 web sites ([www.sanook.com](http://www.sanook.com), [www.kapook.com](http://www.kapook.com) and [www.pantip.com](http://www.pantip.com)) were selected as Thai target social media.

TABLE III. EXAMPLE OF KEYWORD IN THAI

#	Keywords in Thai	Translation in English
1	สึนามิ 2554	Tsunami 2011
2	สึนามิ and 2554	Tsunami and 2011
3	สึนามิญี่ปุ่น 2554	Tsunami in Japan/ Japan sunami 2011
4	สึนามิญี่ปุ่น and 2554	Tsunami in Japan and 2011
5	สึนามิญี่ปุ่น 2554 not สึนามิ	Tsunami in Japan and 2011
6	สึนามิญี่ปุ่น 2554 or แผ่นดินไหวญี่ปุ่น 2554	Tsunami in Japan 2011 or Japanese earthquake 2011/ Earthquake in Japan 2011
7	สึนามิ 2011	Tsunami 2011
8	สึนามิ and 2011	Tsunami and 2011
9	สึนามิญี่ปุ่น 2011	Tsunami in Japan/ Japan Tsunami 2011
10	สึนามิญี่ปุ่น and 2011	Tsunami in Japan and 2011
11	สึนามิญี่ปุ่น 2011 not สึนามิ	Tsunami in Japan 2011 no Tsunami
12	สึนามิญี่ปุ่น 2011 or แผ่นดินไหวญี่ปุ่น 2011	Tsunami in Japan 2011 or Japanese earthquake 2011/ Earthquake in Japan 2011
13	แผ่นดินไหวที่ญี่ปุ่นปี 2554	Earthquake in Japan 2011
14	แผ่นดินไหวที่ญี่ปุ่น and ปี 2554	Earthquake in Japan and 2011
15	แผ่นดินไหวญี่ปุ่น 2554 not แผ่นดินไหว	Earthquake in Japan and 2011
16	แผ่นดินไหวที่ญี่ปุ่นปี 2011	Earthquake in Japan 2011
17	แผ่นดินไหวที่ญี่ปุ่น and ปี 2011	Earthquake in Japan and 2011
18	สารกัมมันตรังสี	Radioactive elements
19	8.9ริกเตอร์	8.9 Richer
20	สึนามิ	Tsunami
21	แผ่นดินไหว	Earthquake

### 3. Thai Social Media Analysis Using Our Method

We applied STEP A and STEP B of our method to 3 Thai social media selected in Section 2.

#### 3.1. STEP A: Crawling social media messages and extracting significant terms

##### 3.1.1. Data Crawling

We tried to crawl messages on 3 web sites (www.sanook.com, www.kapook.com and www.pantip.com) selected which are the highest references of news/blog/web board related

to the Great East Japan Earthquake according to the first study. We crawled these 3 web sites related to the East Japan Great Earthquake with the keywords specified in Section 2, on the dates from March 11, 2011 - June 10, 2011 for 3-month period. As similar topics, there are more than 2,000 messages: kapook 1339 messages, sanook 465 messages and pantip 397 messages as shown in Table IV. Those messages were easily crawled, not biased, and have enough number of authors. There are more than 20,000 words, more than 7 million MB of text.

TABLE IV. CRAWLING DATA: # OF MESSAGES BY WEB SITE AND DATE

Site	Kapook	Sanook	Pantip
Date	messages	messages	messages
11 Mar. 2011	207	195	218
12 Mar. 2011	330	69	124
13 Mar. 2011	130	41	44
14 Mar. 2011	159	29	10
15 Mar. 2011	170	54	
16 Mar. 2011	61	22	1
17 Mar. 2011	84	14	
18 Mar. 2011	46	11	
19 Mar. 2011	48	6	
20 Mar. 2011	13	3	
21 Mar. 2011	8	7	
22 Mar. 2011	12	1	
23 Mar. 2011	9	8	
24 Mar. 2011	15	4	
25 Mar. 2011	4		
26 Mar. 2011	5		
27 Mar. 2011	4		
28 Mar. 2011	6		
29 Mar. 2011	2	1	
30 Mar. 2011	2		
31 Mar. 2011	2		
1 Apr. 2011	2		
2 Apr. 2011	1		
3 Apr. 2011	2		
4 Apr. 2011	1		
5 Apr. 2011	0		
6 Apr. 2011	0		
7 Apr. 2011	4		
8 Apr. 2011	11		
9 Apr. 2011	1		
Total	1339	465	397

### 3.1.2. Morphological Analysis for Thai Language

Then we did the morphological analysis for crawled messages.

Unlike Japanese language, Thai language is a tonal language, which means that the same word can convey different meanings depending on the tone with which it is pronounced. Problems of the Thai Language are[11]:

- A) There are no articles (a, an, the).
- B) No modification or conjugation for tenses, plural, gender, or subject-verb agreement.
- C) Most of the time, question words come at the end of a sentence.
- D) It is a tonal language, which means that the same word can convey different meanings depending on the tone with which it is pronounced. These tones are mid, low, falling, high and rising.
- E) Thai is devoid of inflection (such as the rising voice an English speaker might make to show that he is asking a question). Instead, mood, questions, negation, and other parts of speech are constructed by adding certain words to sentences.
- F) The expression of numbers is shown with separate numerals, quantifiers, and when counting, classifiers.
- G) The uses of particles, which are untranslatable words added to the end of a sentence to indicate politeness, respect, a request, encouragement or other moods
- H) It is an isolated language that writing continuously. There are no space between words and a delimiter for indicating the word boundary is not explicitly used in a Thai text.
- I) The ambiguity problem due to Thai writing system exists, such as the word ตากลม [tak lom] that means “blow wind” or [taa khlom] that means “an round eye.” The character “น (k)” it can be final consonant of the first word or it can be initial consonant of the second word. This ambiguity of the meaning is caused by the character’s position.

Due to above problems, it is not easy to extract keywords from Thai language using morphological technique. To evaluate the effectiveness of morphological technique for Thai language, we did the followings:

1. Use the Swath program[12] to do word segmentation
2. Sort the words according to the statistics and remove functional words and analyze content words
3. Identify keywords from content words from data

Here, the Swath (Smart Word Analysis for Thai) is word segmentation software for Thai[12]. It is an open source program and can be used freely. Fig. 3 shows an example of word segmentation by Swath.



กระทู้ | pantip|.com |  
 1. | สีนามี | 2554 |  
 1.1 | บทบาทหน้าที่ของสถานเอกอัครราชทูตและสถานกงสุลไทยในต่างประเทศ  
 ซึ่งจำกันได้ถึงเหตุการณ์แผ่นดินไหวและสึนามิเมื่อเดือนมีนาคม | 2554 | ที่ผ่านมา  
 ได้สร้างความสูญเสียครั้งใหญ่กับชาวญี่ปุ่นในหลายครั้ง  
 หากแต่ใครจะรู้บ้างกับธรรมชาติครั้งนี้ นำมาซึ่งรอยน้ำตาให้แก่ครอบครัวเล็กๆ ของหญิงไทยคนหนึ่ง  
 โดยได้พาทัวร์ชีวิตสามีชาวญี่ปุ่นของเธอไปด้วย ก่อนหน้าที่จะเกิดเหตุการณ์อันไม่คาดฝันนี้  
 หญิงไทยคนดังกล่าวมีชีวิตครอบครัวที่สมบูรณ์กับสามีชาวญี่ปุ่น เธอตั้งครรถ์และมีอาการแพ้ท้องอย่างรุนแรง  
 จึงต้องกลับมาก่อนอยู่ที่เมืองไทย แต่เธอก็ยังคงติดต่อไปมาหาสู่กับสามีอยู่เสมอ  
 จนกระทั่งเธอให้กำเนิดบุตรชาย สามีได้เดินทางมาเยี่ยมเธอและลูกที่ประเทศไทย  
 และซึ่งวางแผนที่จะมารับเธอและลูกกลับไปอยู่ที่ประเทศญี่ปุ่นด้วยกันอีกครึ่ง แต่โชคชะดากลับเล่นตลก  
 เพียงหนึ่งเดือนก่อนที่เขาจะมารับเธอที่ประเทศไทย สีนามีได้พาทัวร์ชีวิตเขาไปอย่างไม่มีการกลับ  
 ที่จึงให้เธอและลูกน้อยเผชิญชะตา

Fig. 3 An example of word segmentation for Thai language using Swath

Using Swath, it is not easy to solve above problems regarding word segmentation in Thai. So, we've corrected errors manually for keywords automatically extracted. Table V shows some example of words extracted from 3 websites. There are both single and compound words and both concrete and abstract words, such as body, do, water as single words, tall building, World War II, global warming as compound word, plant, people, weapon as concrete words, and frightened, is in trouble, power as abstract words. Selected words are shown in Table V.

TABLE V. EXAMPLE OF WORDS FROM WEB SITES.

kapook.com		sanook.com		pantip.com	
กรรม	karma	ก้ม	bend down	1	One
กระจก	glass	กระทำ	perform	11	eleven
กลัว	frightened	ก๊อ๊ก	tap	12	twelve
กัมตรังสี	radioactivity	กาย	body, physical	กำลัง	Force, power
การใช้ชีวิต	life style	กำลังใจ	mental support	เกิด	Was born, happen, take place
ก้าวหน้า	progress	กิน	eat, consume	ใกล้	Close
กำลังใจ	support	เกียรติ	honor	ขนาด	size
เขื่อน	dam	คน	people	ขอ	Ask
โชคดี	good luck	คนแก่	elderly people	ของ	Thing, stuff, of
คน	person	คิด	think	ข่าว	news
ความรัก	love	คุกคาม	threaten	ขึ้น	Go up, increase, rise

In addition, there are many words expressing deep meaning related Thai culture or characteristic or belief, for example; “karma” (กรรม, บาปกรรม), “mindful” (มีสติ, สติ), “pray”(สวด)[express belief of Buddhism] “misfortune” (เคราะห์) “good luck” (โชคดี) “misfortune” (เหตุร้าย)[express belief of fortune] “feel sorry or feel pitiful” (สงสาร), “don’t give up” (สู้), “kindness” (น้ำใจ), “respect” (นอบน้อม), “help” (ช่วย, ช่วยเหลือ), “donate”(บริจาค) [express kind character] Moreover, a lot of words discussed about Japan and Thai people express good relation between Japan.

In addition, in SNS messages, we found the phrase “11 มีนาคม 2554” frequently. Thai people use Buddhist Era Calendar which is different from Gregorian Calendar. From Thai members, we found that the date meant 11th March 2011. To analyze the SNS messages, cultural knowledge like this is also required so that Japanese could understand what Thai people write and post.

In this research, we’ve discussed and compared the research data translated to English words, because Japanese members cannot read Thai and Thai members cannot read Japanese.

TABLE VI. THAI WORD AND ITS MEANINGS.

Thai word	phonetic	meaning	feature
บาน	/baan/	bloom	main word
บ้าน	/baan^/	house	tone changed
จาน	/jaan/	plate	consonant changed
บิน	/bin/	fly	vowel changed

### 3.2. STEP B: Constructing co-occurrence networks

Then, we’ve formed the Graphs for confirming the contents of messages. Fig.4, Fig.5 and Fig.6 show the network graphs of kapook.com, sanook.com and pantip.com respectively. In Fig.4, Fig. 5 and Fig. 6, there are words related to the nuclear such as “radio activity,” “nuclear,” and “nuclear reactor.” There are also supportive words such as “worry,” “support,” “Don’t give up,” “help,” and “love,” and words about earthquake damage such as “natural disaster,” “tsunami,” and “shake.” These words show that there are topics for the nuclear plant accident, afflicted people and the severe earthquake

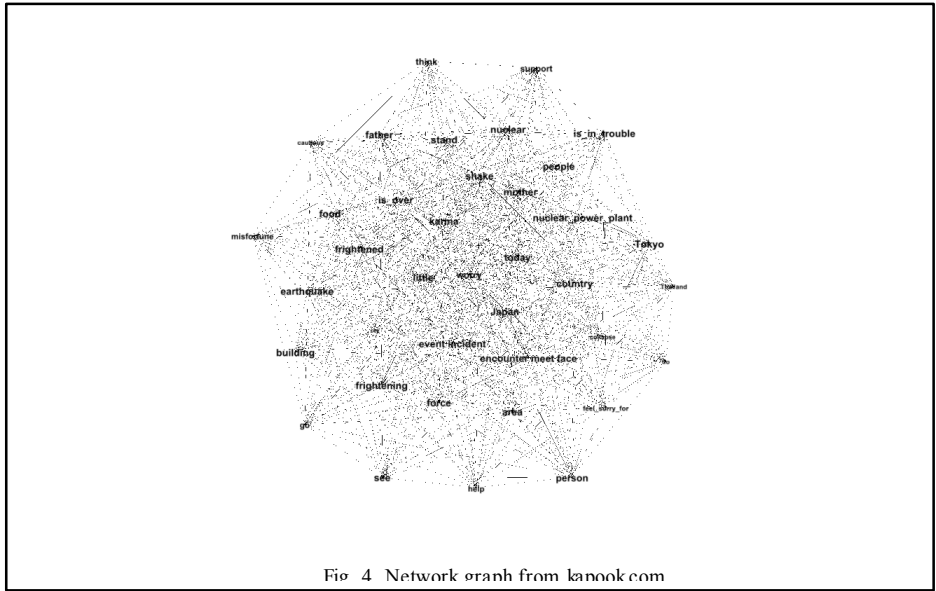


Fig. 4 Network graph from kanook.com

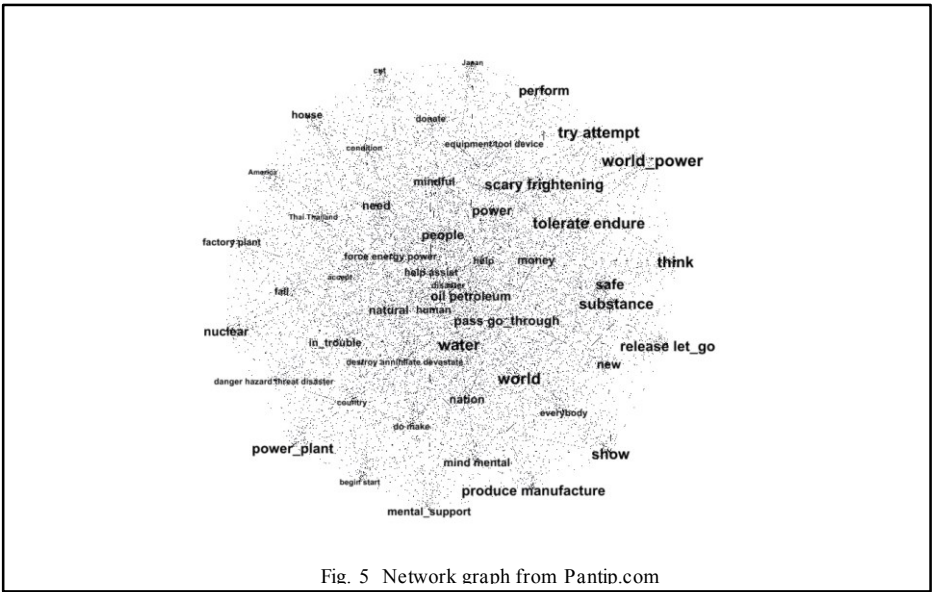


Fig. 5 Network graph from Pantip.com



- T-1. Topic about the nuclear plant accident,
- T-2. Topic about afflicted people support
- T-3. Topic about the severe earthquake.

The relationships between the above 3 social media in Thai and Japan are shown in Fig. 6. K-3 and T-2 are topics for supporting afflicted people so that they can be recognized reactions for afflicted people needs (B-1 and B-2). K-1 and T-3 are similar topics, so that it shows that even in Thai, people also discussed damage by the earthquake. K-2 is about the electricity problem that happened immediately after the earthquake in Japan. It was also important topic at that time.

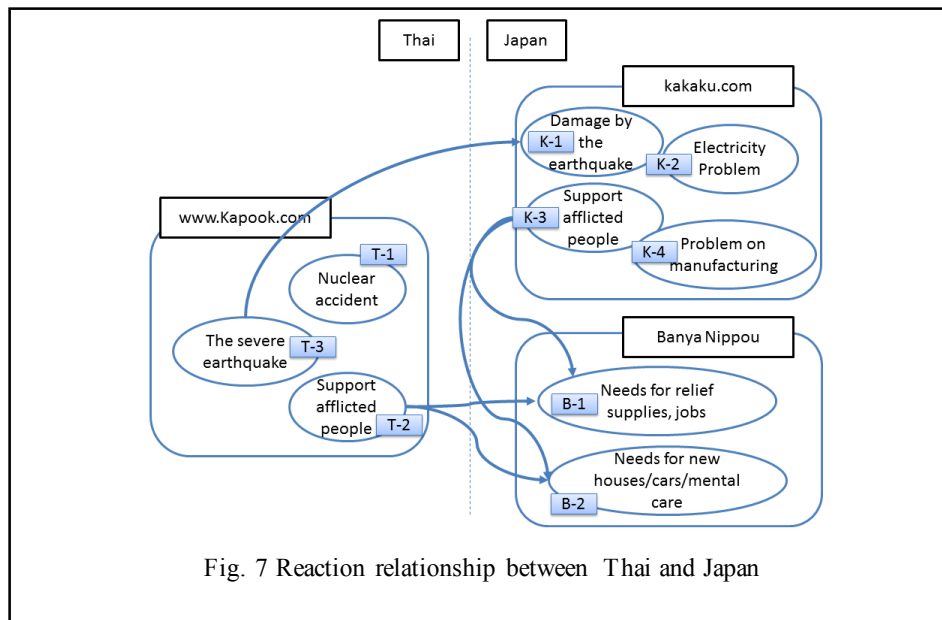
In addition, in kapook.com, we could observe as follows:

- Immediately after the earthquake, people mainly discussed the damages by the earthquake.
- As time went along, the topic for supporting affected people in Japan was emerging.

This means that after the disaster, people are surprised at its serious damage, and then recognize the importance for supporting affected people.

## 5. Related Work

Most related works for detecting topics focus on single media such as blogs, twitter, and web videos respectively. Sekiguchi et al. [13] treated recent blogger posts and



analyzed the word co-occurrence and the repeating rate of word. They visualized the

relation between words and showed topics in social media through the visualization results. Asur et al. [14] investigated trending topics on Twitter. They proposed a simple model based on the number of tweets and found that the resonance of the content with the users of the social network plays a major role in causing trends. Liu et al. [15] and Cao et al. [16] focus on web video analysis. Especially, Cao et al. [16] clusters video tags into groups to get small events and then link these events into topics based on textual and video similarity. On the other hand, our proposed method focuses on multiple social media and analyzes them. It can flexibly show concepts transition by taking into cross-media over time. As for cross-media analysis, most existing works focus on co-clustering among multiple social media. Xue et al. [17] proposed the cross-media topic detection method that was based on co-clustering and detect new topics. Our proposed method focuses on keywords extracted from social media and then detect co-occurrence patterns among them that can be recognized as topics. Because our method is based on keywords, it can be language independent.

Regarding research on detecting temporal relations, Radinsky et al. [18] proposed Temporal Semantic Analysis (TSA), a semantic relatedness model, that captures the words' temporal information. They targeted words in news archives (New York Times, etc.) and used the dynamic time warping technique to compute a semantic relation between pre-defined words. Wang et al. [19] proposed time series analysis which has been used to detect similar topic patterns. They focus on specific burst topic patterns in coordinated text streams and try to find similar topics. Zhou et al. [20] addressed the community discovery problem in a temporal heterogeneous social network of published documents over time. They showed temporal communities by threading the statically derived communities in consecutive time periods using a new graph partitioning algorithm. Qiu et al. [21] focused on the problem of discovering the temporal organizational structure from a dynamic social network using a hierarchical community model. The above existing methods focused on single media and analyzed their transition. In our method, on the other hand, other languages' social media is targeted. Our research final goal is to develop language-independent social media analysis environment.

## 6. Conclusion

This paper described the preliminary approximation to apply our social media analysis method to Thai language. Compare with morphological analysis of Japanese, the morphological analysis of Thai words is much more difficult as we mentioned in the paper. Especially the analysis of Thai SNS messages is more difficult than ordinary Thai texts because the latest words are more likely to appear. In a sense, this paper can be a pioneer work of multi-language message analysis. The feature of our SNS message analysis method is that we can use the same method to any language messages. Basically we are using the same method to both Thai and Japanese.

In our application, social media candidates in Thai were selected through our examination. Retrieval keywords for crawling social media messages were also adjusted by counting the number of messages that have corresponding keywords. Then, SWATH, that is the morphological analysis software for Thai Language, was adopted to extract words form messages. We discussed morphological analysis errors caused by

characteristics of Thai language. We also analyzed how Thai people reacted to the earthquake by comparing reactions on social media between in Thailand and in Japan.

In this paper, we applied STEP A and STEP B in our method. It is a future work to apply other STEPs (STEP C, and D) to Thai social media as well and analyze Thai people reaction over time. Then we will compare reactions between Thai and Japan over time more precisely. Finally, we plan to develop the language independent social media analysis platform for analyzing different reactions in cross-cultural environment.

## References

- [1] T. Hashimoto, T. Kuboyama, B. Chakraborty, Y. Shirota, Discovering Topic Transition about the East Japan Great Earthquake in Dynamic Social Media, *GHTC 2012* (2012), 259–264.
- [2] S. Higuchi, T. Hashimoto, T. Kuboyama, K. Hirata, Exploring Social Context from Buzz Marketing Site - Community Mapping Based on Tree Edit Distance -, *Proc. of PerCol 2013 (Fourth International Workshop on Pervasive Collaboration and Social Networking)* (2013), pp.187-192.
- [3] T. Hashimoto, S. Aramvith, T. Chauksuvanit and Y. Shirota, Comparison of Reaction in Social Media after the East Japan Great Earthquake between Thailand and Japan, *Prof. of ISCIT2013 (International Symposium on Communications and Information Technologies)* (2013), 781- 786.
- [4] SAVE IWATE (Non-profit organization), *Banya Nippo*, [http://sviwate.wordpress.com/in\\_english/](http://sviwate.wordpress.com/in_english/).
- [5] K. W. Church, W. A. Gale, Poisson mixtures, *Natural Language Engineering 1* (1995), 163–190.
- [6] M. E. J. Newman, Modularity and community structure in networks, *Proc. of National Academy of Science USA 103(23)*(2006), 8577–8696.
- [7] T. K. Landauer, S. T. Dumais, A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge, *Psychological Review*, 104(2) (1997), 211-240.
- [8] T. Hashimoto, B. Chakraborty, T. Kuboyama, Y. Shirota, Temporal Awareness of Needs after East Japan Great Earthquake using Latent Semantic Analysis, *Proc. of EJC2013 (23rd European-Japanese Conference on Information Modelling and Knowledge Bases)* (2013), 214-226.
- [9] B.W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, *Biochim. Biophys. Acta*, 405 (1975), 442–451.
- [10] Thai Internet Innovation Research Institute, *Top Website Award Truehits.net Web Award 2011* (2012), <http://truehits.net/> [2012, July 24].
- [11] Basic Introduction to Thai Language, [http://www.peacecorps.gov/wws/multimedia/language/transcripts/TH\\_Thai\\_Language\\_Lessons.pdf](http://www.peacecorps.gov/wws/multimedia/language/transcripts/TH_Thai_Language_Lessons.pdf).
- [12] Software: SWATH - Thai Word Segmentation, <http://www.cs.cmu.edu/~paisarn/software.html>
- [13] Y. Sekiguchi, H. Kawashima, T. Uchiyama, Discovery of related topics using series of blogsites' entries, *JSAI 2008, 211-1* (2008) (in Japanese).
- [14] S. Asur, B. A. Huberman, G. Szabó, C. Wang, Trends in social media: Persistence and decay, *ICWSM 2011* (2011), 434-437.
- [15] L. Liu, L. Sun, Y. Rui, Y. Shi, S. Yang, Web video topic discovery and tracking via bipartite graph reinforcement model, *IWWW 2008* (2008), 1009-1018.
- [16] J. Cao, C. W. Ngo, Y. D. Zhang, J. T. Li, Tracking web video topics: discovery, visualization and monitoring, *IEEE Transactions on Circuits and Systems for Video Technology*, 21(12) (2011), 1835-1846.
- [17] Z. Xue, Z. Jiang, G. Li, Q. Huang, Cross-media topic detection associated with hot search queries, *ICIMCS '13*(2013), 403-406.
- [18] K. Radinsky, E. Agichtein, E. Gabrilovich, S. Markovitch, A word at a time: Computing word relatedness using temporal semantic analysis, *WWW 2011* (2011), 337-346.
- [19] X. Wang, C. Zhai, X. Hu, R. Sproat, Mining correlated bursty topic patterns from coordinated text streams, *KDD 2007* (2007), 784-793.
- [20] D. Zhou, I. Council, H. Zha, C. L. Giles, Discovering temporal communities from social network documents, *ICDM 2007* (2007), 745-750.
- [21] J. Qiu, Z. Lin, C. Tang, S. Qiao, Discovering organizational structure in dynamic social network, *ICDM 2009* (2009), 932-937.

# Evaluation of A Flipped Classroom & Analysis of Students' Learning Situation in A Computer-Programming Course

Yasuhiro HAYASHI, Ken-ichi FUKAMACHI and Hiroshi KOMATSUGAWA  
*Faculty of Photonics Science, Chitose Institute of Science and Technology*

**Abstract.** We have put into practice the flipped classroom as a way of utilizing e-learning contents in our computer-programming course since 2013. The main feature of this practice is to use most of the time usually dedicated to lecture for practicing by assigning the students the e-learning materials as preparation before the class. We gave the students homework to learn vocabularies and grammar of the C programming language. This decreased the time a teacher spent lecturing and the students were assigned applied-problems to make practical software in addition to conventional basic problems in training. Our goal is to maintain the students' motivation to learn the computer programming through the sense of accomplishment that each student obtains by finishing practical assignments in the training. We confirmed the effectiveness of this approach by comparing examination scores between last year and this year, and putting questionnaires to the students. Additionally, we analyzed the learning situation of the students who were weak in programming. The results are shown in this paper.

**Keywords.** Computer Programming, Flipped Classroom, e-learning, ICT, Blended Learning

## 1. Introduction

It's useful that students learn computer programming as a way to cultivate abilities of logically thinking and solving many social problems while the society has dynamically changed by Information-Communication Technology (ICT). Many classes for the computer programming in ordinary universities consist of lecture and training and a lot of time for them are spent. However, many students feel that it's difficult to make a code. Even if the students understand grammar of a computer programming language, they can hardly make a code within the training. Thus, there are some students who don't work on exercises until a model answer is shown and also a few students who copy the code written by friends might appear. In a state of affairs like this, the class time isn't taken advantage of effectively. The teaching method of computer programming is required to improve the abilities and the motivation of each student.

The flipped-classroom inverts traditional teaching methods, delivering instruction online outside of class and moving "homework" into the classroom [2, 9, 10] by utilizing learning materials of massive online open courses (MOOCs) [3, 4, 8]. As the results of legacy research on the e-learning system, Moodle that is developed in open-source community for educational systems has been widespread in the world [5], and SCORM [6, 7] is provided as a set of technical standards and specifications for web-



based e-learning software. Although we can study through massive e-learning contents anytime and anywhere, it is important whether each student's ability really improves by utilizing the system and learning materials. In the flipped-classroom, the learning for preparation by the e-learning system is needed. Each student studies the learning contents through the provided online homework in advance, and can clarify difficult points by themselves. Additionally, the teachers can get learning histories of each student by confirming the learning management system (LMS) on the e-learning system, and based on them, the teachers can make a plan of the lecture before the class.

In the spring term of 2013, we carried out a class of the computer programming that applied the flipped-classroom by the e-learning system that has been developed in our university. The class that is named 'Programming Skill' handles C programming language. The main feature of this approach is to use most of the time usually dedicated to lecture for practicing by assigning the students the e-learning materials as preparation before the class. We gave the students homework to study vocabularies and grammar of the C language. This decreased the time a teacher spent lecturing and the students were assigned applied-problems to make practical software in addition to conventional basic problems in training. Our goal is to maintain the students' motivation to learn the computer programming through the sense of accomplishment that each student obtains by finishing practical assignments in the training.

## 2. Practicing of A Flipped Classroom in A Computer-Programming Course

We have practiced improvement of computer programming courses in our university by using an e-learning system every year in order to develop students having information-communication technology. In this time, the flipped-classroom was adopted among a course that taught second grade students the C programming language in three required courses. The learning contents in this course were a review of fundamental grammar that the students learned in the first grade and the advanced grammar such as functions, pointers and structures. After that, the students learn the object-oriented programming using Java in autumn semester of the second grade.

■ 合計・平均  
配列と繰り返し文を組み合わせることで、合計・平均など統計処理を行うことができる。

```
#include <stdio.h>
#define N 10

main() {
    int i, sum;
    float ave;
    int a[N];

    for (i = 0; i < N; i++){
        printf("Input a[%d]", i);
        scanf("%d", &a[i]);
    }

    sum = 0;
    for (i = 0; i < N; i++){
        sum = sum + a[i];
    }

    ave = (float)sum / (float)N;

    printf("sum=%d\n", sum);
    printf("ave=%f\n", ave);
}
```

各要素にキーボードから値を代入する。

sum は 0 に代入する。  
もし 0 にしないと、  
次の for 文において、 $i=0$  のとき、  
未定の sum と  $a[i]$  を足すことになり、  
正しい計算ができなくなる。

(古い) sum と  $a[i]$  を足し算し、  
その結果を (新しい) sum に代入する。

合計を個数で割り算し、平均を求める。  
平均値は必ず実数値となるため、  
sum, i と実数型に変換 (キャスト) する。

もしキャストしない場合、  
int 型 / int 型の計算と見なされ、  
割り算の結果は int となり、端数は切り捨てられる。

10	i
584	sum
58.4	ave

配列 a

a[0]	65
a[1]	52
a[2]	80
a[3]	71
a[4]	40
a[5]	92
a[6]	44
a[7]	38
a[8]	76
a[9]	26

ページ 1

Figure 1. A Screen Chapter of the e-Learning Contents as The Textbook

The e-learning system that was used for this approach has developed in our university since 2000 and it's a semi-open courseware. Its service has been provided to

many universities, high schools and junior high schools in Japan. Many students have used it for free and the whole user accounts are over 40,000. The materials of various fields such as mathematics, physics and information technology have been set up and have been managed by teachers and students of the participating institutions. Each material consists of textbooks and exercises. A screen capture of a textbook is shown in figure 1. The textbooks include explanations, figures and animations corresponding to codes for the students' easy understanding and are also used as presentation sheets in the class. In the exercises, answers that each student responds through a web based input form are automatically marked. A series of learning histories of each student is recorded to the learning management system and the teachers can check the data in detail such as the learning time, the number of times of watching textbooks, the number of times of watching hits, the exercises results, achievement degree of each homework.

In this approach, the course was conducted according to schedule shown in Table 1 and is the same as an average year. For introduction of the flipped-classroom, we told the students the following explanation:

1. Do a homework before the lesson of every week
2. Read the textbooks, and clarify the easy and difficult points for you
3. Answer questions of the exercises as much as possible
4. The teachers confirm the learning history of doing the homework.

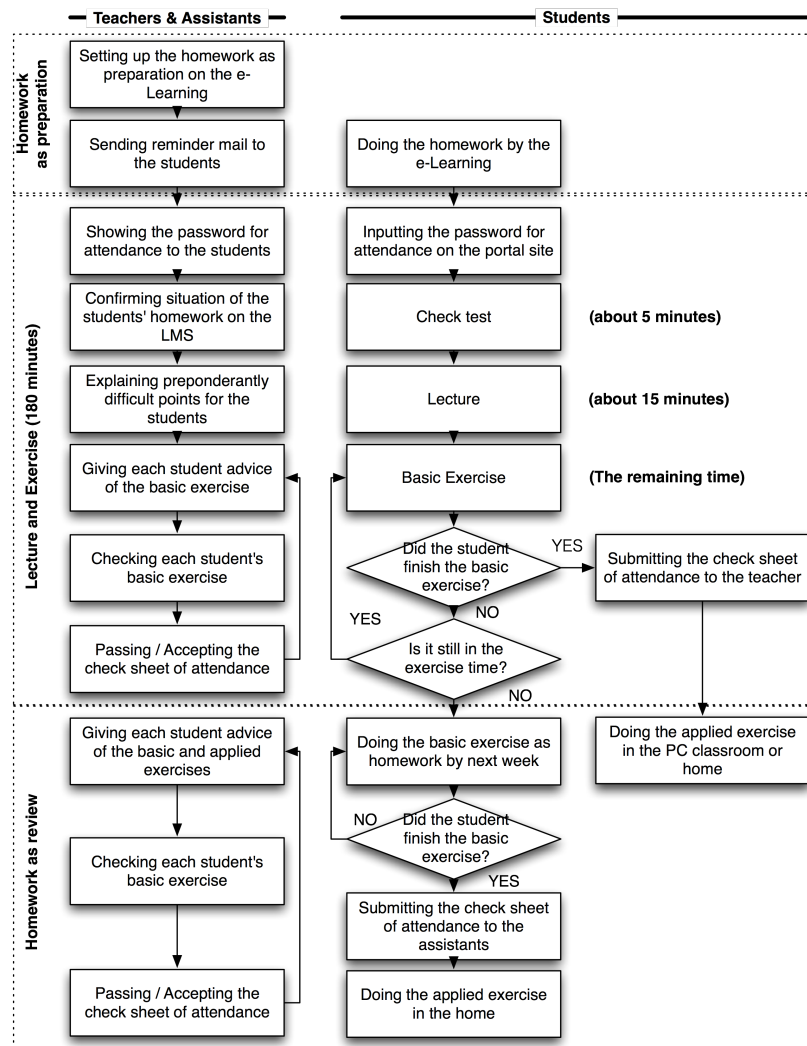
The fundamental grammar of C language that the students had already learned in the first grade was assigned as the homework from the 1st to the 4th week. An important point is that these reviews were the homework for the flipped classroom in order to ease the students' worry about changing teaching style to the flipped-classroom. The advanced grammar of C language that the students newly learned was deployed from the 5th to the 13th week. A final assignment to make a code of the Whack-A-Mole game was conducted at the 14th and 15th weeks. A game player inputs a coordinate of moles that appear on the screen at random and strikes them. After showing the students a requirement of the game, the teachers evaluated whether the code made by each student meet the requirement and also conducted an oral examination to check whether they can explain detail of the code. The mid-term and the final examination were conducted at the 8th and the 16th weeks in order to confirm the students' degree of comprehension. Both examinations consist of two types, a writing exam for checking the programming skill and a web-based exam for the knowledge of programming.

**Table 1.** The learning contents and schedule of 'Programming Skill' as a subject

Week	Learning Contents	Week	Learning Contents
1st	Guidance, Review (UNIX, Variables)	9th	Pointer 1: Fundamental
2nd	If Statement	10th	Pointer 2: Array and Pointer
3rd	For Statement	11th	Pointer 3: Function and Pointer
4th	1-Dimensional Array and 2-Dimensional Arrays	12th	Structure 1: Fundamental
5th	Function 1: Fundamental	13th	Structure 2: Structure and Array
6th	Function 2: Use of Global Variables	14th	Final assignment
7th	Function 3: Use of Libraries	15th	
8th	Mid-exam	16th	Final-exam

The course flow is shown in Figure 2. Before a weekly class, the teachers provide all students homework through the e-learning system, and encourage them by a reminder mail. The students do the homework and can ask teaching assistants questions.

In the class, a web-based check test that consists of several questions is done for about 10 minutes in order to clarify difficult points that the students feel in the homework. While the check test, the teachers confirm the learning histories and find students who did not do the homework. Finishing the test, the lecture and the training are begun.



**Figure 2.** The flow of study of the flipped-classroom

In the lecture, the teachers avoid to explain the grammar of C language because the lecture is premised on the homework. On the other hand, they easily explain contents that students felt difficult and the instruction that is required for the training. In order to maintain the students' concentration, an explanation is performed for 15 or fewer minutes and the long explanation is divided into two parts, and the easy training is set up in between. Operating computers is not allowed to the students in order to prevent missing hearing it while the explanation. By this improvement, the lecture became shorter for about 10 minutes than an average year. Moreover, the students go into the

exercises consisting of a basic exercise and an applied exercise for acquiring the grammar and a series of processes of practical software development through a rock-paper-scissors game programming. However, the students who did not do the homework do the lecture after finishing the homework in the class. A handout that is described a procedure for making a code and algorithms is also provided to them in order to supplement the e-learning materials and the teacher's explanation.

The students' attendance requires two verifications. Each student inputs a password through the portal site at the time of the lecture start in order to prevent the students coming in late and also an attendance sheet for the is collected by the teaching assistants after finishing the exercises. The students who completed all exercises within the exercise time can leave the PC classroom because the teachers and the assistants can intensively support the other students. If the students cannot complete the applied exercise by the end of the class, they can make it homework that they should complete until next week. The assistants check it again and then, the sheet is accepted.

### 3. Evaluating the Effectiveness of the Flipped Classroom & Analyzing the Students' Learning Situation

Using the same questions of 2012 in the mid-term and final exams in order to confirm the effectiveness of this approach, we compared the examination scores between 2012 and 2013. Number of the students in 2012 was 57 who were science and engineering major and in 2013 was 88. 3 teachers and 8 teaching assistants performed this class adopting the flipped classroom. The area covered in the mid-term exam was from the 1st to the 7th weeks and the final exam was the entire weeks.

**Table 2.** The Results of mid-exam in 2012 and 2013

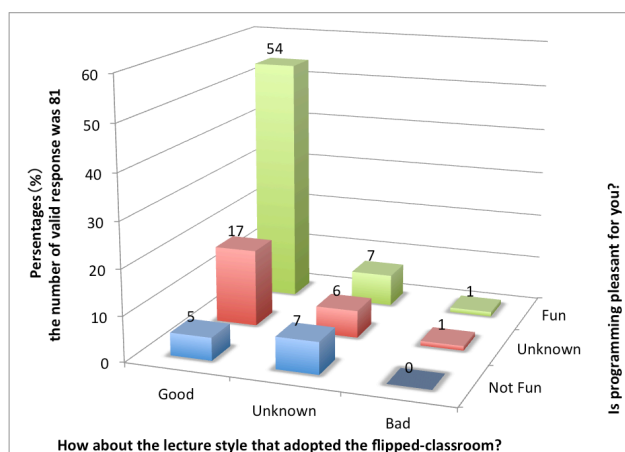
Year	Web-based Test		Written Test	
	2012	2013	2012	2013
Max Score	100	100	97	98
Min Score	24	48	0	21
Average Score	79.3	86.9	66.3	70.1
Standard Deviation	14.6	11.6	18.2	17.7

**Table 3.** The Results of final-exam in 2012 and 2013

Year	Web-based Test		Written Test		
	2012	2013	2012	2013 *	
Max Score	98	98	4	4	95
Min Score	7	24(48)	0	0	0
Average Score	69.6	78	0.94	1.87	60.4
Standard Deviation	17.2	16.8	1.36	1.15	25.0

The results of the mid-term and final exams are shown in table 2 and 3. Compared with 2012, the average scores of the web-based test and the written test on the mid-term exam became high, and the minimum scores of both tests also became high. The standard deviation of both tests became low and the minimum score of the web-based test of the final exam was 48 excepting a student who attended at only the exam without attending at the class. Moreover, only one question was set as the writing test of the final exam of 2012 and was scored out of four points. Eight questions including the question were set in 2013\*. The score of only the same question of 2012 (the left side) and the score of the writing test (the right side) are shown in the column of 2013

in Table 3. We consider that effectiveness is shown in the flipped-classroom in those students who achieved a mid to high-level grade on examinations through the results.



**Figure 3.** The students' feelings about the flipped-classroom and the computer programming

We examined the students' feelings about their programming skill through questionnaires to all students, and valid responses were 81 students. The results became as follows: "I can write a program very much." was 7 students. "I can write a program." was 58. "I cannot write a program well." was 12. "I cannot write a program." was 4. And also, we also examined the students' feelings about the flipped-classroom and the computer programming through this approach. The result is shown as Figure 3. 54% answered that the flipped classroom was good and the computer programming was fun. Furthermore, we got following positive comments from many students: "I have understood the learning contents by the flipped classroom". On the one hand, we also got following negative comments: "When I forget homework, it becomes difficult for me to catch up in the lecture", "I need to spend time in order to do the assignment alone in the exercise time. It's a loneliness". Finally, we got following teachers' comments: "Compared with an average year, the students seems to be able to write a code", "The students who understood about the difficult Pointer seem to be many."

Furthermore, examining the learning situation of the students in detail from the learning histories, we considered a way of improving the flipped classroom and analyzed the learning situation of each student, categorizing superior and inferior groups on the basis of 60 percent of the total scores because there was no correlation seen between the homework time and the examination scores. The superior group was 69 students and the inferior group was 19. The total score goes up to five hundred points because the score is summation of all exams including the final assignment. 60% of the scores are standards to grant a credit to the students. The key-statistics data of both groups are shown in Table 4 and 5. The total learning time means summation of doing homework. The score of the mid-term exam that consists of the web-based test and the written test goes up to two hundred points as summation. As a point to note, we have realized that the students (No. 1-8, 13-15) where the learning time is longer than the students of the superior group are in the inferior group. The learning time of the students of the cell that the background color on Table 5 has reversed is longer than the average of the superior group. They must have studied hard, however the result is not

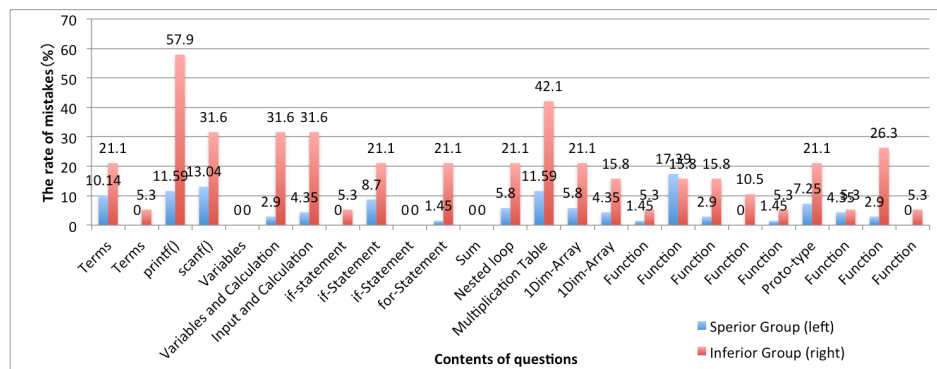
brought to them. On the other hand, the students (No. 16-19) seem to have given up studying after the mid-term exam.

**Table 4.** The key-statistics data of the superior group

	The total learning time by the mid-term exam	The total learning time between the mid-term and the final exam	The score of the mid-term exam	The final score
Average	451.6	280.1	168.3	219.2
Maximum	1227	999	198	278
Minimum	77	0	116	140
Standard Deviation	247.1	191.8	16.3	34.5

**Table 5.** The key-statistics data of the inferior group

Student No.	The total learning time by the mid-term exam	The total learning time between the mid-term and the final exam	The score of the mid-term exam	The final score
1	425	505	130	168
2	432	407	77	139
3	539	233	118	127
4	256	343	112	146
5	474	193	97	128
6	550	576	136	148
7	937	526	106	55
8	1024	633	110	118
9	384	323	122	109
10	268	190	146	141
11	383	265	116	114
12	256	164	104	137
13	453	173	136	154
14	355	389	140	154
15	419	294	103	134
16	252	75	132	148
17	149	52	112	120
18	434	128	124	148
19	244	37	83	140
Average	433.4	289.8	116	133.1
Maximum	1024	633	146	168
Minimum	149	37	77	55
Standard Deviation	215.3	173.9	18.2	23.6



**Figure 4.** The rate of mistakes of each question by web-based text on the mid-term exam

Additionally, we calculated the rate of mistakes on the web test of the mid-term exam in order to find out difficult points of the inferior group and the result is shown in figure 4. The exam included the contents that the students had already learned at the first grade. The highest rate of mistakes was “printf()” that shows letters and variable values on a screen. “Multiplication table” that uses nested loop processing was second higher. And, third was “scanf()” that inputs numerical values to the variables by a user’s action. It seems that the flipped classroom wasn’t effective to the inferior group because the flipped classroom greatly relies on the students’ motivation.

#### 4. Conclusion and Future Works

We have shown the results of the practical approach of the flipped classroom for the computer programming of our university in this paper. We got knowledge about putting the flipped classroom into practice. (1) Effectiveness was shown in the flipped classroom in those students who achieved a mid to high-level grade on examinations by comparing examination scores between 2012 and 2013. (2) 54% of the students answered that the flipped classroom was good and the computer programming was fun in the questionnaires. (3) We have realized that the several students where the learning time is longer than the students of the mid to high-level grade are in the inferior group. However the good result is not brought to them.

As the future works, we should carry out a pre-test to be conducted in the guidance in order to find out the students who might become the low-level grade in the early stage of the class. Moreover, simple exams should be conducted in order to confirm the learning situation of each student in detail. The visualization of the learning histories and the teaching team for data analysis are required.

#### References

- [1] Harumasa TADA, Hiroyuki MARUTA: "The Course Design with Repetitive Lessons for Education of Programming," Bulletin of Kyoto University of Education. No.116, 2010.
- [2] Baker, J.W.: "The 'classroom flip': Using web course management tools to become the guide by the side." Paper presented at the 11th International Conference on College Teaching and Learning, 2000.
- [3] Fred G. Martin: "Will massive open online courses change how we teach?" Communications of the ACM, Volume 55 Issue 8, Pages 26-28, August 2012.
- [4] Kerry Wu: "Academic libraries in the age of MOOCs," Reference Services Review, Vol. 41 Iss: 3, pp.576 – 587.
- [5] Dougiamas, M. & Taylor, P. (2003). Moodle: Using Learning Communities to Create an Open Source Course Management System. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003 (pp. 171-178).
- [6] Qu, Changtao, and Wolfgang Nejd. "Towards interoperability and reusability of learning resources: A SCORM-conformant courseware for computer science education." Proc. of the 2nd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2002), Kazan, Tatarstan, Russia. 2002.
- [7] Bohl, Oliver, et al. "The sharable content object reference model (SCORM)-a critical review." Computers in Education, 2002. Proceedings. International Conference on. IEEE, 2002.
- [8] Liyanagunawardena, Tharindu Rekha, Andrew Alexandar Adams, and Shirley Ann Williams. "MOOCs: A Systematic Study of the Published Literature 2008-2012." International Review of Research in Open & Distance Learning 14.3 (2013).
- [9] Ashish Amresh: Adam R. Carberry, John Femiani: "Evaluating the effectiveness of flipped classrooms for teaching CS1," Frontiers in Education Conference, IEEE, pp. 733-735, 2013.
- [10] Herreid, Clyde Freeman, and Nancy A. Schiller. "Case studies and the flipped classroom." Journal of College Science Teaching 42.5 (2013): 62-66.

# Time - A Multidimensional Concept

Anneli HEIMBÜRGER  
*anneli.a.heimburger@jyu.fi*  
 University of Jyväskylä  
 Faculty of Information Technology  
 Finland

**Abstract.** There exist a lot of studies about time, its interpretations, different features and structures from several scientific points of view. In our paper, we propose a multidimensional framework of time. The main idea of the paper is to present a synthesis of different dimensions of time. We discuss some parts of the framework to illustrate and highlight the multidimensional features of time. We also demonstrate an early-stage implementation of the framework as a "Time on Wall" course in the eEducation/Optima environment. By means of the "Time on Wall", we are able to teach different dimensions of time across disciplines and faculties and to illustrate different time scales.

**Keywords.** Time, multidimensional framework of time, models of time, cultural sense of time, eEducation, Optima environment

## Introduction

*"Living on Earth may be expensive, but it includes an annual free trip around the Sun." Anonymous in Singh, S. 2005. Big Bang: The Origin of the Universe.*

When we travel to a different country, we assume that a certain amount of cultural adjustment will be required, whether it's getting used to new food or negotiating a foreign language, or adapting to a different standard of living or another currency. We have to adapt to another culture's sense of time and the pace of life which both contributes also to our sense of disorientation. For example in Brazil, it is perfectly acceptable to be three hours late, and in Japan we can find a sense of the long-term that is often unheard of in the Western countries. Time is everywhere around us and, in a way, inside us.

Over the last two thousand years, philosophers have been interested in "What is time?" Different philosophical time conceptions have been proposed changing each other [1]. The concept of time is of great interest not only to philosophy but also to science. Humankind regards time as a universal phenomenon. We try to understand our world and the universe by means of time. It is difficult to find an object which wouldn't have a relation to time. Research of any process has a temporal context. Time has special characteristics, such as a rhythm and scale in each of these processes. Time is a term that aggregates temporal properties of our world. In the nature, several time scales exist, starting from the macro level - the estimated age of the universe ( $13.8 \times 10^9$  years or  $4.4 \times 10^{17}$  seconds) - and reaching to the very micro level in particle physics - Planck's time ( $5.4 \times 10^{-44}$  seconds) [2]. In today's science we already are dealing with the concept of time at nanoscale in nature. These scales are very difficult for us to



understand. We should be able to illustrate them, however. In science, several time categories have been proposed: for example, physical, geological or biological time [3].

The evolution of information and communication technologies has given us further extensions to the temporal context, such as time in databases, time in Web and XML applications and time in mobile computing. As time is an important parameter of all processes in nature, science, society and in our technological systems as well, we should be able to teach time.

Even if it is difficult to give an exact and right answer to a question “What is time?”, we can still study time from different viewpoints. In our paper, we integrate different aspects of time into a global view and propose a multidimensional framework of time, which is presented in Figure 1. Our paper discusses some parts of the framework and highlights the multidimensionality of time by illustrating some multidimensional features of time. Finally, we demonstrate the early stage of the implementation of our framework, the “Time on Wall” course in the eEducation/Optima environment. “Time on Wall” is a multidisciplinary course on time based on the proposed framework. By means of the “Time on Wall”, we are able to teach different dimensions of time and to illustrate different time scales across disciplines and faculties.

The rest of the paper is organized as follows. Section 1 provides different definitions of time. Models of time are presented in Section 2. In Section 3, cultural dimensions of time are discussed. Time in information systems is discussed in Section 4 and in other disciplines in Section 5. The early stage of implementation of our framework is introduced in Section 6. Section 7 is reserved for the conclusion and issues for further research.

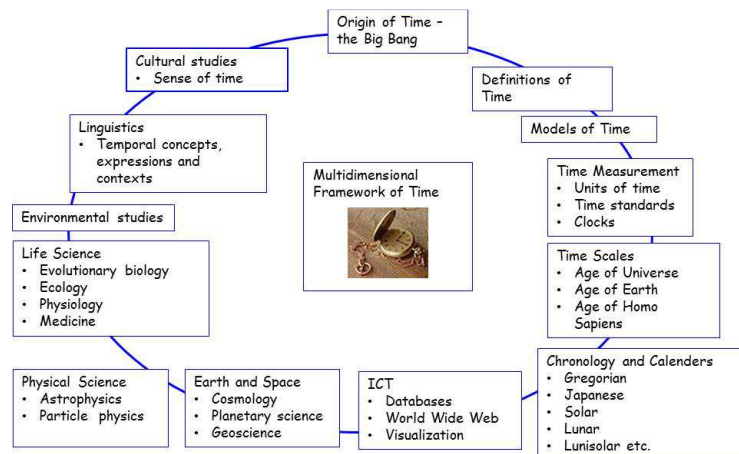


Figure 1. A multidimensional framework of time.

## 1. Definitions of Time

What is time? Several definitions can be found in the literature. Time is a measured or measurable period, a continuum that lacks spatial dimensions [4]. Time is the indefinite

continued progress of existence and events in the past, present and future, regarded as a whole [5]. It is an observed phenomenon, by means of which human beings sense and record changes in the environment and in the universe [6]. Time is a basic component of the measuring system used to sequence events, to compare the durations of events and the intervals between them, and to quantify the motions of objects [7].

Time is a dimension in which we order events from the past through the present into the future. We can measure durations of events and intervals between them. We usually understand the passing of time by means of changes we observe occurring to objects in space, like their transformations over time and their movements in relation to one another. Scientific study of the spatial processes that have an effect on these changes is impossible without considering both space and time. Only by this way we are able to derive cause and effect relationships and reach ultimate understanding of nature and its structure. We can say that time has a structure as well as primitives. As a whole, time seems to be one of the ways we humans try to analyze and understand our world and universe. On Earth, the Sun has an essential role in our sense of time.

Most of these definitions mainly reflect the characteristic way humans understand time in terms of events. According to Giumale and Kahn [8], time, as an abstract concept, means a space of time points connected to each other by *before* and *after* -like operators. Some features in the structure of time reported by Schreiber [9] include absolute ordering (past, present and future), relative ordering (before, concurrent-with, after), finiteness and infinity, openness and closure, discreteness and continuity, objectivity and subjectivity, and linearity and circularity.

Temporal logic is any system of rules and symbolism for representing, and reasoning about, propositions qualified in terms of time. In the field of temporal logics, several researchers have presented the notion of branching time [10, 11]. In this notion, the model of time is a tree-like structure in which the future is not determined. There are different paths in the future, any one of which might be an actual path that is realized. It is used in formal verification of software or hardware artifacts.

Granularity, in general, is the extent to which a system is broken down into small parts, either the system itself or its description or observation [9]. It is the extent to which a larger entity is subdivided. For example, a day includes 24 hours, an hour 60 minutes and a minute 60 seconds.

Time can also be represented in terms of time primitives, that is, by durations, duration bounds, time points, time intervals, concurrency, coincidence, synchronicity and periodicity [12]. Duration is the absolute distance between two points in time. It is specified in terms of years, months, weeks, days, hours, minutes and seconds. Duration boundaries are defined by an upper and lower boundary such as a minimum duration of 2 minutes and a maximum duration of 20 minutes. Time points are used to represent specific instants along a timeline. Time intervals are sets of constraints between two points, a start and an end time. Concurrency is the closeness of two or more temporal events in time, in no particular order. Coincidence describes the intersection of several intervals. Synchronicity is the synchronous occurrence of two temporal events. Periodicity is the repetition of the same event with a constant period.

Many famous philosophers have argued over two contradictory aspects on time [2]. One aspect – the objective view - is that time is part of the fundamental structure of the universe in which events occur in sequence. Sir Isaac Newton supported this view, and hence it is sometimes referred to as Newtonian time. The other aspect is that time does not refer to any kind of "container" that events and objects "move through", nor to any

entity that "flows". Instead, time is part of a fundamental intellectual structure within which humans sequence and compare events – the subjective view.

In the International System of Units [13, 14], time is one of the seven fundamental physical quantities. Time is used to define other quantities such as velocity. Therefore, defining time in terms of such quantities would result in circularity of definition. An operational definition of time, wherein one says that observing a certain number of repetitions of one or another standard cyclical event (such as the passage of a free-swinging pendulum) constitutes one standard unit such as the second - the base unit of time. This definition is useful for both advanced experiments and everyday life.

Temporal measurement has occupied scientists and technologists. It also was a prime motivation in navigation and astronomy. Periodic events and periodic motion have long served as standards for units of time. Examples include the apparent motion of the sun across the sky, the phases of the moon, the swing of a pendulum and the beat of a heart. Currently, the international unit of time, the second, is defined in terms of radiation emitted by cesium atoms [14]. Time has also significant social importance. It has economic value and personal value, because we are aware of the limited time in each day and in human lifespans.

## 2. Models of Time

Modeling of time has two main traditions represented in the literature. One view of time is a set of points without duration [15]. In time-point data models, observations are associated with a specific point in time. The other model proposes that intervals should be considered as temporal individuals [16, 17].

According to Allen's interpretation [16], an interval is an undefined basic concept the meaning of which derives from the relations in which it stands with other intervals - relations such as "overlaps", "contains", "comes before" etc. Allen's temporal relationships between two time objects, X and Y, are given in Table 1.

**Table 1.** Allen's temporal relationships between two time objects X and Y.

Relations	Example	Inverse
X before Y	XXX YYY	YYY XXX
X equals Y	XXX YYY	YYY XXX
X meets Y	XXXXYY	YYYYXXX
X overlaps Y	XXXX YYYY	YYYY XXXX
X during Y	XXX YYYYYYY	YYY XXXXXXXX
X starts with Y	XXX YYYYYYY	YYY XXXXXXXX
X ends with Y	XXX YYYYY	YYY XXXXX

With temporal data models, one of the classic questions is how temporal data is represented. In time-point data models, observations are associated with a specific point in time. The most commonly employed concept is order or concurrency. In time-

interval data models, observations are associated with the time between two time points. Most models focus on three temporal concepts: order, concurrency, and synchronicity. Time interval models are summarized in Table 2 [12, 15].

**Table 2.** Time interval models.

Time Interval Models	Definitions
Allen's interval relations	Thirteen relations, forming an algebra. Any two intervals have exactly one of the relations. Invented in AI for temporal ... <i>Continues</i> ...reasoning. Used also in data mining.
Freksa's semi-interval relations	Semi-interval means that one interval boundary is unknown. Two relations between start or end points of two intervals suffice to uniquely identify the relations. Representation of incomplete or coarse situations is easier than with Allen's relations.
Reich's interval/point relations	Extension of Allens's relations to points. Five more relations: point finishes and inverse, point starts and inverse, point equals.
Roddick's mid-point interval relations	Allen's relation extended by a relation of each interval midpoints to the other interval. For example: midpoints within other intervals (largely overlap) or not (overlap to some extent). Nine versions of overlaps, 49 relations in total.

### 3. Time in Information Systems

Basic models of time have been applied to information systems in several ways. In the open specifications developed by the W3C community, time can be found at least in the following specifications: Working with Time Zones [18], XML/HTML [19], EmotionML [20] and Time in OWL [21]. Working with Time Zones describes the guidelines and best practices for working well with geographically distributed applications with date and time values. The document also aims to provide a basic understanding and vocabulary for talking about time and time handling in software, a source of confusion for many developers and content authors on the Web. In HTML, the <time> element represents either a time on a 24 hour clock, or a precise date in the calendar, optionally with a time and a time-zone offset. In Emotion Markup Language, time is represented as four timestamps: absolute time, duration, relative time, and timing in media. Time in OWL presents an ontology of temporal concepts. The ontology provides a vocabulary for expressing facts about topological relations among instants and intervals, together with information about durations, date times and time zones. The main classes of the ontology are: TemporalEntity (subclasses: Instant and Interval), DurationDescription, DateTimeDescription, TemporalUnit, and DayOfWeek. Time in OWL has been proposed to be extended with the concept of temporal aggregates [22]. Temporal aggregates are collections of temporal entities. Examples of temporal aggregates are "every 3<sup>rd</sup> Thursday in 2008", and "3 consecutive Mondays".

XLinkTime [23, 25] is a time-sensitive linking structure and an extension of XML Linking Language (XLink) [25]. XLinkTime consists of resources and/or portions of resources and links between them. The links are functions of time, and they can be activated by a user when a certain temporal rule or a set of temporal rules is valid.

XLinkTime is realized by defining a timerule namespace `xmlns:timerule` with attributes `timerule:start`, `timerule:end`, `timerule:status` and `timerule:title`.

Temporal entities in content level are significant issues for temporal reasoning [26]. Temporal expressions can be explicit, implicit or relative. They can also include uncertainty of temporal durations. Examples of explicit temporal expressions are the token sequences “January 2014” or “September 14, 2014, 3.00 p.m.”. Implicit temporal expressions include the token sequence “Ocean Day 2014 in Japan”, which can be mapped to the expression “July 21, 2014”, or the sequence “Midsummer Day 2014 in Finland”, which can be mapped to “June 21, 2014”. Implicit temporal expressions can also be collections of temporal entities such as “every other Wednesday in every second month”. Relative temporal expressions represent temporal entities that can only be anchored in a timeline in reference to another explicit or implicit, already anchored temporal expression (for example the starting time of a meeting). For example, the expression “3 p.m.” alone cannot be anchored in any timeline. However, it can be anchored if the date of the meeting is known. The date then can be used as a reference for that expression, which then can be mapped to a timeline. Uncertainty of temporal durations is interesting. Consider a news article title *Prof. A met with Prof. B in Tokyo*. How long did the meeting last? Our first inclination is to say we have no idea. But in fact we do have some idea. We know the meeting lasted more than ten seconds and less than one year. By guessing and narrowing the bounds, our chances of being correct will increase. Just how accurate can we make duration judgments like this? Will it be possible to extract this kind of information from text automatically? The uncertainty of temporal durations has been recognized as one of the significant issues for temporal reasoning.

A temporal database is a database with built-in supports for handling data involving time, for example a temporal data model or a temporal version of Structured Query Language (SQL) [27]. More specifically, the temporal aspects usually include *valid time* and *transaction time*. These attributes can be combined to form bitemporal data. Valid time is the time period during which a fact is true with respect to the real world. Transaction time is the time period during which a fact stored in the database is considered to be true. Bitemporal data combines both valid and transaction time. It is possible to have timelines other than valid time and transaction time, decision time in a database, for example. In that case, the database is called a multitemporal database as opposed to a bitemporal database.

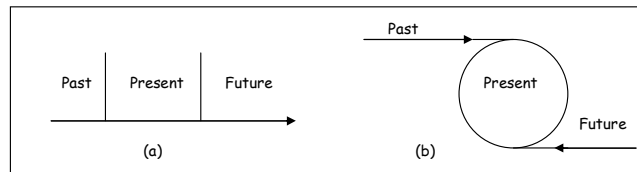
The field of temporal data mining studies has ordered data streams with temporal dependencies and interdependencies. Over the last decade, many interesting techniques of temporal data mining have been proposed and shown to be useful in many applications [28, 29].

According to Zhezhnych and Peleschychyn [30], we can summarize different time dimensions in a context of information systems as follows: (1) time has an ordering feature, (2) time is discrete, (3) time orders unique individualized events, (4) in time, events are divided into past, present and future ones, (5) time flows (future events become present, events of present become past etc.), (6) time is universal, (7) time is irreversible, (8) alternative scenarios of future events are possible, but only one scenario will be realized and (9) time has a meta-moment structure (every present corresponds to its past and to a set of possible futures).

#### 4. Cultural Dimensions of Time

Time is seen in a different way by eastern and western cultures, and even within these groupings temporal culture differs from country to country. Also temporal identities of different organizations and teams in organizations may vary. In cultural context, there exist two general time models: linear and cyclic [31]. In the linear time model (Figure 1a), past time is over, present time can be seized and parceled and made to work for the immediate future. One task is carried out at a time. For example, Scandinavian people are essentially linear-active, time-dominated and monochronic. They prefer to do one thing at a time, concentrate on it and do it within a scheduled timetable. Southern Europeans are more multi-active and polychronic. Monochronic cultures differ from polychronic cultures in that the former encourage a highly structured, time-ordered approach to life and the latter a more flexible, indirect approach, based more upon personal relationships than scheduled commitments.

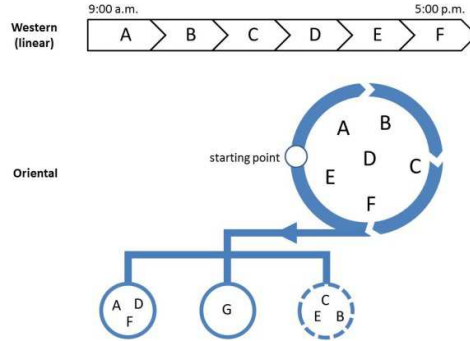
Cyclic time sees time as circular, not necessarily leading towards something but repeating itself in a cycle of events. Each day the sun rises and sets, the seasons follow one another, people grow old and die, but their children reconstitute the process. In many Asian countries, time has traditionally been considered as cyclic. For example, the Japanese traditional temporal culture can be presented by the Makimono model of time (Figure 2) [7]. In Makimono time, the future flows into the present, just as the past does. The present is a period that links the region of the past with the world of the future. Nowadays, linear time model has also been integrated into Japanese society. At present, Japan uses the Gregorian calendar, together with year designations stating the year of the reign of the current emperor [32].



**Figure 2.** Linear time model (a) and cyclic time model according to Makimono time pattern (b). Makimono takes its name from *makimono*, a picture story or writing mounted on paper and usually rolled into a scroll.

Linear time is believed to be the closest depiction of our experience of time. However, for many centuries the Eastern world has used a cyclical view of time which supports the way nature behaves.

Western linear type action chains and Asian reflection are compared in Figure 3 [31]. The western model contains tasks A-F to be sequentially completed during the day. In Asian reflection, instead of tackling problems immediately in sequential fashion, to circle round them for some time is preferred. After a suitable period of reflection, A, D and F may seem worth pursuing. B, C and E may be quietly dropped. Contemplation of the whole scene has indicated however that task G might be the most significant of all.



**Figure 3.** Western action chains versus Asian reflection.

When travelling, we are not only moving over time zones but also moving to a different kind of temporal culture. We can move in the opposite temporal direction, from fast to slow (or from slow to fast) [33]. From a temporal point of view, there are several things to be taken into account. We can consider these as temporal rules or temporal norms in cultural context (Table 3).

**Table 3.** Examples of temporal rules in cultural context.

Temporal expressions	Time has several meanings in different language. For example in Japanese, which is a very semantic, context sensitive and context rich language, there are around nine different time expressions that can be used in different contexts.
Punctuality	We should learn to translate – by cultural mapping of time frames from our own culture to another – our appointment time to the accepted range of punctuality for a particular situation in another culture.
The line between work time and social time	We should understand the separating line between work time and social time. In Japan, the distinction between work and social time can be meaningless. There the workday has a large social element and social time is very much a part of the work. The crucial goal that overrides both of these types of time is the <i>wa</i> of the work group. <i>Wa</i> is a Japanese cultural concept usually translated into English as "harmony". It implies a peaceful unity and conformity within a social group, in which members prefer the maintenance of a harmonious community over their personal interests.
Waiting for another person	We should also study the rules of the waiting game: who is expected to wait for whom and for how long?
Spaces between events	How your hosts treat pauses, silences or doing nothing at all. For Japanese people, the spaces between events are as significant as the events themselves: for example, the length of time of a silence that must be endured before a "yes" means "no".
Asking about accepted sequences	Each culture sets rules about the sequences of events. Is it work before play or vice versa? Do people take all sleep at night, or is there a siesta in the mid-afternoon? Is one expected to have coffee or tea and socialize before getting down to serious business, and if so, for how long? Etc. <i>Continues...</i>

Clock time or event time	Are people on clock time or event time? In monochronic cultures, one activity is scheduled at a time; in polychronic cultures, people prefer to switch back and forth from one activity to another in a very flexible way.
Practice	An intellectual understanding of temporal norms does not in itself insure a successful transition. Practice is needed.
Criticism	We should not criticize what we do not understand. This concerns also our temporal norms. We can always ask our host to explain.

To summarize the main difference between western and eastern sense of time, we can say that the western time orientation emphasizes objectivity, absoluteness and fixation of time. The Asian traditional cultural time orientation conceives time as subjective, relative and flexible. The understanding of a cultural concept of time is an issue for successful cross-cultural communication and cross-cultural collaboration.

## 5. Examples of Time Dimensions in Natural Sciences

Many scientists, including Galileo and Newton, up until the 20th century thought that time was the same for everyone everywhere. In classical, non-relativistic physics, time is a scalar quantity and, like length, mass, and charge, is usually described as a fundamental quantity. Our modern conception of time is based on Einstein's theory of relativity, in which rates of time run differently, depending on relative motion, and space and time are merged into space-time. We live on a world line rather than a timeline. The world line of an object is the unique path of that object as it travels through 4-dimensional space-time. Thus time is a part of a coordinate, in this view.

Astrophysicists believe the entire Universe and therefore time itself began about  $13.8 \times 10^9$  years ago in the Big Bang. The Big Bang theory [34] is the prevailing cosmological model that describes the early development of the Universe. The age of the Earth is calculated to be  $4.54 \times 10^9$  years. Homo sapiens originated in Africa about 200,000 years ago.

In biology, evolution is any change across successive generations in the heritable characteristics of biological populations. Evolutionary processes give rise to diversity at every level of biological organization. The similarities between all present-day organisms indicate the presence of a common ancestor from which all known species, living and extinct, have diverged through the process of evolution.

Chronobiology is a field of biology that examines periodic (cyclic) phenomena in living organisms and their adaptation to solar- and lunar-related rhythms. These cycles are known as biological rhythms. The variations of the timing and duration of biological activity in living organisms occur in many essential biological processes. The most important rhythm in chronobiology is the circadian rhythm, a roughly 24-hour cycle shown by physiological processes in all these organisms.

Environmental science provides an integrated, quantitative, and interdisciplinary approach to the study of environmental systems [35]. Environmental issues almost always include an interaction of physical, chemical, and biological processes. The key characteristics of an effective environmental scientist include the ability to relate space and time relationships as well as quantitative analysis. In natural disasters resulting from the Earth's natural processes, including floods, volcanic eruptions, earthquakes and tsunamis, time has an essential role from short-term, alarm scale to long-term scale



for estimating environmental effects locally and globally. Time geography was originally developed by human geographers. Hägerstrand's earliest time geography formulation that uses a physical approach informally described the workings of large socio-environmental mechanisms [36]. Hägerstrand's approach involved the study of how events occur in a time-space framework, and he illustrated it by means of a graphical notation. Today, time geography is applied in multiple fields related for example to transportation, regional planning, geography, time use research, environmental science and virtual spaces [37]. Time geography is an evolving multidisciplinary perspective on spatial and temporal processes and events such as social interaction, ecological interaction, social change and environmental change. Time geography is an integrative ontological framework and visual language in which space and time are basic dimensions of analysis of dynamic processes.

## **6. How to Teach Time? - Implementation in Optima Environment**

Imagine that the entire history of the universe is compressed into one year - with the Big Bang corresponding to the first second of the New Year's Day and the present time to the last second of December 31st (midnight). Using this scale of time, each month would equal a little over a billion years. By means of this scaling and our Time on Wall -system, we can illustrate the times of occurrence of important events in our imaginary one-year universe. The same kind of scaling can be realized at a micro level, in particle physics and cell biology, for example.

Our early stage implementation of the multidimensional framework of time is realized in the Optima system [38]. The Optima system is widely used in universities in Finland as a Web-based learning environment. The University of Jyväskylä is divided into 15 individual Optima environments, each faculty having its individual Optima environment. Each of these environments has hundreds of workspaces. If a user has access to more than one Optima environment, he/she can change the environment easily. It is easy to create courses across faculties. Technical support for Optima is working well, and the system contains built-in tools and functions for e-learning. Wiki environments are also of considerable interest in this context. However, they need more customization for our purpose.

The users of workspaces are divided by the status of their profile: supervisors (teachers, lecturers and owners of the workspace) and users (students). Both profiles have different permissions and actions available. The profiles are workspace-specific: a person can be a supervisor in one workspace and a user in another.

A material repository is a variant of workspace. Workspaces are recommended to be set as material repositories when the same material is to be used in multiple different workspaces. Materials are linked to different workspaces from the originals which reside in a material repository. Using a material repository has its benefits: when there is a need to update some material, only the material in the material repository needs to be updated - workspaces that have the materials linked in them are automatically updated from the source.

A workspace is a closed working area created for a single online course. Access to it is always restricted. Material is normally organized in folders. In a workspace, various tasks such as distributing course materials, group work, discussions, completing assignments etc. can be performed. Objects are documents and functions (e.g., a page made with a light web editor, a discussion list, a link, an imported pdf file

and a return box) in the workspace. These objects can be organized into folders. The screen print of our preliminary version of the Time on Wall course implementation is shown in Figure 4.

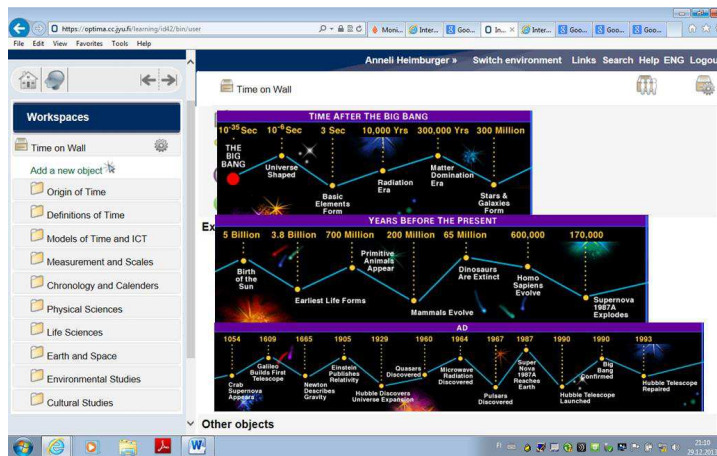


Figure 4. “Time on Wall” course prototype in Optima environment.

## 7. Conclusions

In our paper, we have studied time from different viewpoints and integrated different aspects of time into a global view. We proposed a multidimensional framework of time and demonstrated an early-stage implementation of the framework, the “Time on Wall” course in the eEducation/Optima environment. By means of the “Time on Wall”, we are able to teach different dimensions of time and to illustrate different time scales.

Time is a “connecting language” between different scientific disciplines. For the next phase of our research, we will organize a multidisciplinary scientific seminar around the concept of time. Its main aim is to specify the requirements for the final design of the Time on Wall course before establishing it as a part of international activities of the eEducation/Digital Campus programme at the Faculty of Information Technology of the University of Jyväskylä.

## References

- [1] Markosian, N., *Time*. *Stanford Encyclopedia in Philosophy* referred Dec 13, 2013 <URL: <http://plato.stanford.edu/entries/time/#TopTim>>.
- [2] Planck ESA, Planck 2013 results XVI. Cosmological parameters, submitted to *Journal of Astronomy and Astrophysics*. Dec 13, 2013 <URL: <http://arxiv.org/abs/1303.5076>>.
- [3] Birx, H. J. (ed.), *Encyclopedia of Time: Science, Philosophy, Theology, and Culture*, Sage, Thousand Oaks, CA, 2009.
- [4] *Encyclopedia Britannica Online*, referred Dec 13, 2013 <URL: <http://www.britannica.com/>>.
- [5] *The Compact Oxford English Dictionary*, referred Dec 13, 2013 <URL: <http://www.oxforddictionaries.com/>>.
- [6] *Whatis.com*, referred Dec 13, 2013 <URL: <http://whatis.techtarget.com/>>.

- [7] *Wikipedia: Time*, referred Dec 13, 2013 <URL: <http://en.wikipedia.org/wiki/Time>>.
- [8] Giumale, C. A. and Kahn, H. J., An information model of time. In: *Proceedings of the 30th International Annual ACM IEEE Design Automation Conference*, ACM Press, New York, NY, USA, 668 – 672, 1993.
- [9] Schreiber, F. A., Is time a real time? An overview of time ontology in informatics. In: Halang, W. A. and Stoyenko, A. D. (Eds.) *Real Time Computing*, Proceedings of the NATO Advanced Study Institute, Sint Maarten, Dutch Antilles, 5 – 17 October, 1992. Springer-Verlag, Berlin, Germany, 283 – 307.
- [10] Emerson, E. A. and Halpern, J. Y. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *Journal of Computer and Systems Sciences* **30**(1985), 1-24.
- [11] Ben-Ari, M., Pnueli, A. and Manna, Z. The temporal logic of branching time. *Acta Informatica* **20**(1983), 207-226.
- [12] Moerchen, F., Temporal Pattern Mining for Time Points, Time Intervals, and Semi-Intervals, *The Eleventh SIAM International Conference on Data Mining*, Mesa, Arizona, April 28 - 30, 2011.
- [13] International Bureau of Weights and Measures, *The International System of Units (SI)* (8th ed.), ISBN 92-822-2213-6, 2006.
- [14] Unit of time (second), *SI Brochure*. BIPM, 2012.
- [15] Van Benthem, J., *The logic of time. A model-theoretic investigation into the varieties of temporal ontology and temporal discourse*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 300 p, 1991
- [16] Allen, J. F., Time and time again: The many ways to represent time. *International Journal of Intelligent Systems* **6**(1991), 341 –355.
- [17] Hirsch, R., Relation Algebras of Intervals. *Artificial Intelligence* **83**(1996), 267 – 295.
- [18] *Working with Time Zones*, referred Dec 13, 2013 <URL: <http://www.w3.org/TR/timezone/>>.
- [19] *XML/HTML*, referred Dec 13, 2013 <URL: <http://www.w3.org/wiki/HTML>>.
- [20] *EmotionML*, referred Dec 13, 2013 <URL: <http://www.w3.org/TR/emotionml>>.
- [21] *Time Ontology in OWL*, referred Dec 13, 2013 <URL: <http://www.w3.org/TR/owl-time/>>.
- [22] Pan, F., A Temporal Aggregates Ontology in OWL for the Semantic Web. In *Proceedings of the AAAI Fall Symposium on Agents and the Semantic Web*, Arlington, Virginia, 2005.
- [23] Heimbürger, A. et al., Time Contexts in Document-Driven Projects on the Web: From Time-Sensitive Links towards an Ontology of Time. In: Duzi, M., Jaakkola, H., Kiyoki, Y. and Kangassalo, H. (eds.) *Frontiers in Artificial Intelligence and Applications, Vol. 154, Information Modelling and Knowledge Bases XV*, IOS Press, Amsterdam, 136 – 153, 2007.
- [24] Heimbürger, A., Temporal Information Processing in the Context of Knowledge Cluster Systems. Jaakkola, H. (Ed.) *Selected Topics on Distributed Disaster Management: Towards Collaborative Knowledge Clusters*. Tampere University of Technology, Pori Publication 12, 84-98, 2008.
- [25] *XML Linking Language*, referred Dec 13, 2013 <URL: <http://www.w3.org/TR/xlink11/>>.
- [26] Pan, F., *Representing Complex Temporal Phenomena for the Semantic Web and Natural Language*. University of Southern California. 164 p, 2007.
- [27] *ISO/IEC 9075-1:2011 Information technology -- Database languages -- SQL -- Part 1: Framework (SQL/Framework)*, 2011.
- [28] Mitsa, T. *Temporal Data Mining*, New York, CRC Press, 2010.
- [29] Hsu, W., Lee, M. L. and Wang, J., *Temporal and Spatio-Temporal Data Mining*. Hershey, PA, IGI Publishing, 2008.
- [30] Zhezhnych, P. and Peleschychyn, A. Time Aspects of Information Systems. *CADSM 2007*, February 20-24, 2007, Polyana, Ukraine
- [31] Lewis, R. D., *When Cultures Collide: Leading Across Cultures*, Nicholas Brealey Publishing, Boston MA, USA, 2000.
- [32] *Calendar at Japan-guide.com*, referred Dec 13, 2013 <URL: <http://www.japan-guide.com/e/e2272.html>>.
- [33] Levine, R., *A Geography of Time*, Basic Books, New York, NY, USA, 1997.
- [34] Singh, S., *Big Bang: The Origin of the Universe*, Fourth Estate, New York, NY, 2005.
- [35] Imura, H., *Environmental Systems Studies. A Macroscopic for Understanding and Operating Spaceship Earth*. Springer, Tokyo, 2013.
- [36] Hägerstrand, T., What about People in Regional Science? *Papers of the Regional Science Association*, **24**(1970), 7-24.
- [37] Sui, D., Looking through Hägerstrand's dual vistas: Towards a Unifying Framework for Time Geography, *Journal of Transport Geography*, **23**(2012), 5-16.
- [38] Optima Guide at JYU, referred Dec 29, 2013 <URL: <https://www.jyu.fi/itp/en/optima-guide>>.

# Grounded Multi-Level Computations

Jaak HENNO

*Tallinn University of Technology*

**Abstract.** Development of Biology, Economy, Information Technology, Social studies etc have introduced and acknowledged the understanding, that we are living among Information Processing Systems. We understand rather well computations in man-made devices, especially in computers, and how these computations change the (logical) state of the world (pre- and post-conditions). But we do not have general model of computations which occur in living organisms and its subsystems, in language, is social systems etc

Here is proposed a unified view of computations what occur in different Information Processing Systems (IPS) and clarified notions of Data and Information; the view is based on Entropy.

**Keywords.** Information, Information Processing System, language, emergence, entropy

## Introduction

Konrad Zuse suggested in 1969 in his book 'Rechnender Raum' [1],[2] that the whole universe is based on computation. At the beginning the wild idea was mostly ignored by most physicists. But nobody, any physicist could not indicate flaws in Zuse's arguments. And with advances in biology, with discoveries of information processing in living systems, in business etc the idea started to attract more and more attention, both in popular culture (the cultist film '*The Matrix*!') and also from serious scientists. MIT physicist Seth Lloyd calculated in 2006 in his book "Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos" [3], that the visible universe has so far computed about  $10^{122}$  operations on  $10^{92}$  bits.

But we do not have general model of computations which occur in living organisms and its subsystems, in language, is social systems, in governments, in Cloud and which are possibly super-Turing complete [4], [5].

Tom Stonier proposed [6], that information is a part of the physical universe the same way as matter and energy.

There are (at least) two semantics of the word 'Information'. The abstract: "information arises when a value is chosen from an available value set" [7], [8] and concrete, semantic, evaluated, where information is considered together with information receiver – only receiver can give information some value, meaning. The first is a mathematical concept based on probabilities, in Information Technology (IT) and also here we are interested in information, which has some meaning, can be evaluated.

Information processing in cells, computers, government organisations, societies etc is rather different. Are there some features, which are common to all these Information processing Systems (IPS)?

The situation is becoming all the time more complex and confused because of the constant Information flood and differences in understanding of words Information and Data.

The total amount of digital data created in World is estimated to be 40000 exabytes in 2020 [9] (exabyte =  $10^{18}$  bytes). This is over 300 times more than it was in 2005.

World population in 2020 is estimated to be 7.72 billion or only 1.176 times more than it was in 2005 [10].

Capacity of human brain is estimated ca 2.5 petabytes [11] (petabyte =  $10^{15}$  bytes). If our brain does not change essentially during these 15 years in 2005..2020, the total memory capacity of mankind will be in 2020 also ca 1.185 times more than in 2005.

Percentage of digital data, what the whole mankind could possibly store, will be in 2020 only 0.385 of what it was in 2005. Thus in 2020 whole mankind could possibly store only 0.00004% of the whole digital data generated in our digital world. And if this trend continues, the percentage of digital data what the whole mankind could manage directly will decrease at least two times in every two years [9],[12], since the amount of digital data grows in two years (at least) twice.

Mankind is (generally) rational. We believe, that all things have a purpose. What is the purpose of this flood of data ? Some people call this 'Information Age', 'Age of Big data', but what is the purpose of amounts of data, what we (humans) can not manage? Software is eating the world [13], who will be in control in 2020 - we or our (?) programs ? How much of this data is Information, which has sense for us ?

## 1. Circularity of human (natural) language

Human language is circular. We define concepts using other concepts, and then these other concepts using again the concepts we started with. Consider e.g. definitions of the basic concept of mathematics, 'set' from the Oxford Dictionary [14]:

*set - a group or collection of things that belong together or resemble one another or are usually found together;*

*group - a number of people or things that are located, gathered, or classed together;*

*collection - a group of things or people*

i.e. set *is-a* group || collection *is-a* set || group *is-a* set || set

- we are back where we started.

Mathematicians have studied such circularities carefully and have come to conclusion, that some words/notions/concepts can't be defined using other words/notions/concepts. We have to assume, that we understand the basic notions from our experience, our perceptions of the world, the same way. These worlds/notions/concepts are *grounded* in our perceptions, our experience. In mathematics, it is assumed, that we understand words 'together', 'set', 'element' the same way. But some properties still should be stipulated: a set cannot be an element of itself and the 'set of subsets of empty set' is a meaningless concept.

We understand concepts, which are based on firsthand experience - hunger, joy, pain etc - more or less the same way, although we cannot check, whether our perceptions are exactly the same (is 'your red' the same as 'my red' ?). These concepts are grounded on our interaction with environment. For abstract concepts, which are derived from these basic concepts, similar understanding is obtained with lot of explanations, clarifications, examples and counter-examples. The more abstract a

concept, the less confident we can be about common understanding. Even simple concepts can be rather confusing:

- are penguins birds ? Oxford Dictionary explains this question using 229 words [15];

- what is game and why do we play ? - 1437 words + 3 images + 10 comments [16]

- what is information ? 671 pages [17] (in Volume 1 - volume 2 has not yet appeared)

What is *Information*, what is *Data*, why these concepts are often used together with *Entropy* and also together with *Energy* [18] ?

## 2. What is Information ?

It is commonly agreed that we now live in "Information Age", we have become an "Information Society."; economists state that information is the (main) source of value in a global economy. Information regulates all processes and information processing is the central characteristics of all living beings. Even a simple unicellular organism is a complex and purposefully organized algorithm. Man is the most complex information-processing system existing on earth. By some estimates the human memory stores 2.5 exabytes and total number of bits processed in every second in the human body is  $3.4 \times 10^{19}$ , but it uses only about 20 watts of power [19] – PC needs a million times as much per calculation [20]. None of human-made computing machines approaches the economy of energy of the brain.

So we are the best information processing devices, but in spite of vast number of papers on information the meaning of the word "Information" remains abstract and vague. The situation has not become essentially better from the famous utterance of Wiener: "Information is information, not matter or energy" [ 21 ]. In spite of proliferation of information systems, there are still no generally agreed answers to the questions – What is Information? Has Information natural properties [22] and if so, then what are these properties ?

There is also a confusion with closely connected notions of 'entropy' and (somewhat less) with 'communication'; recently information is also connected with energy.

In order to clear this (somewhat) messy ground it is good to consider historical development of these terms and how they have been used in technology and language, since the troubles started already when two great men, Norbert Wiener and Claude Shannon laid grounds to theories which are based and use these notions.

The word 'information' is derived from the Latin word 'informare' - "give form to", i.e. information is always represented with some form, structure, pattern. But 'representation' of a concept in human mind is not the concept itself [ 23 ]. Representations are the result of grounding our sensory perceptions.

The Oxford Dictionary explains 'information' as '*facts provided or learned about something or someone*'. It is rather difficult to distinguish this explanation from the explanation of word 'data' from the same dictionary: '*facts and statistics collected together for reference or analysis*' - i.e. 'information is facts provided' and 'data is facts collected'. And both of these definitions refer not only to an entity - 'facts', but also to a processes - 'provided, collected'. Information transfer is communication.

Communication was the main topic of Shannon's research. Shannon always considered communication and did not speak about information; he always used notions 'communication, communications channel'. In his ground-breaking paper "A Mathematical Theory of Communication" [24] he explicitly states: "fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point", but "semantic aspects of communication are irrelevant to the engineering problem". Shannon considered abstract messages and communication without any assumptions about their meaning, but concrete information has meaning. A message without meaning is not information.

Although engineers wanted to keep the name 'Information Theory' for technology, for research in radio, electrons, and wire communications, this Shannon's paper started the 'Information Theory' boom in many fields- linguistics, biology, physics. When asked for clarification of the concept 'Information' he explained in 1953 very carefully: "It is hardly to be expected that a single concept of information would satisfactory account for the numerous possible applications of this general field" [25]. But the 'gin was out of bottle' and more and more people from psychology, neurophysiology, management, linguistics, biology etc were considering their field as part of Information Theory, so that confused Shannon was forced to publish in 1956 the memo "The Bandwagon"[26], where he warned: "Workers in other fields should realize that the basic results of the subject [communication channels] are aimed in a very specific direction, a direction that is not necessarily relevant to such fields as psychology, economics, and other social sciences".

While Shannon was considering only abstract communication, transfer of signals (data) in communication channels, Norbert Wiener considered in his ground-breaking opus "Cybernetics or Control and Communication in the Animal and the Machine" [27] concrete, semantic information: how signals are used for control, both of mechanical and biological systems, i.e. how signals influence, control these systems. Wiener considered signals as information. Signals, data become for receiver information if they make sense for receiver and receiver can use them for achieving its goals.

### 3. Information Processing Systems

All Information Processing systems (IPS) - living systems, businesses, languages, computer programs, governments etc - are finite and have *goals*, their purpose is to perform some *actions* aimed to fulfil their goals and survive by constantly processing information about threats and opportunities in the world around them. Goals of living systems are metabolism (getting energy needed for their functioning) and reproduction, goals of businesses - produce some goods and/or services, goals of languages - improve communication of language users, goals of governments - guarantee and improve well-being of citizens etc. Actions of IPS systems are based on information they receive from their environment; these actions allow them survive and develop, become more complex, reduce their inner entropy.

Actions for achieving goals can be performed only if some conditions are true. Conditions can be expressed with predicates, thus we can consider *goals as logical predicates*.

Signals, data become information for an IPS only when they can be used for proving their goals. A goal for a living system can be e.g. an eatable object:

$a \wedge \text{eatable}(a)$

If this becomes true for somebody, he/she eats, i.e. this changes his/her behaviour. To prove a goal predicate IPS collects data from his environment. Data from external environment is encoded as certain patterns - visual, chemical, mechanical, olfactory etc. If some data items allow proving some goal predicates, the behaviour of IPS changes - he performs actions of his goal; this allows to improve its structure, reduce its entropy (and increase entropy of the environment).

*A data item is an Information for an IPS, if it changes the future behaviour of the IPS, i.e. acts for the IPS like a program.*

Different IPS have different goal predicates, thus what is information for someone may not be information for others.

For processing (proving a goal predicate) data should be stored by IPS. All IPS are finite, they tend not to store useless data and if something is stored, but could not be used for a long time, then it is forgotten. According to Lindauer's principle [28] forgetting, erasing memory also increases thermodynamic entropy in the environment [29].

At the time when Wiener did his research this was a novel approach to significance of signals. The main aim of Wiener's research was the development of military targeting and communication technology, but his research and established by him new science, Cybernetics, presented also deeper understanding of life as being at its core information processing. The computers had just arrived and the word 'program' was not yet very well-known, so Wiener explained the concept 'Information' with bold statement: "Information is information, not matter or energy" [21], p. 132.

Many researchers use the word 'information' in a very broad sense: "By information I mean data processing in the broadest sense; the storage, retrieval, and processing of data becomes the essential resource for all economic and social exchanges. These include: data processing of records... data processing for scheduling... data bases" [30]. If all data processing is information, then what is data?

This very broad treatment of information - every number becomes information when it has some attribute, some semantics - may historically be induced by deficient resources for data storing and analysing, so 'just in case' every piece of data was considered valuable. But this is unnecessary, unused data is just forgotten. We learn best when the new information is a logical continuation to what we already know and can be right away used in practical actions [31].

Understanding that data becomes information only when it changes the (future) behaviour of receiver, i.e. works like a program has gradually become accepted by more and more researchers.

M. Burgin defines information as a "... information for a system is a capacity to cause changes in the system" [32, page 99].

K Haefner postulated [33], that all natural systems are Information Processing Systems; each IPS can receive, store, process and transmit information; information processing is an essential internal feature of all systems; the whole universe may be viewed as a gigantic IPS. Information is a system variable and we should distinguish between system's internal information which is an essential component of every natural system and external information, which is communicated between systems and measured by some external measuring system. Physics and biology are interested (mainly) in internal information; Information Technology (IT) – in external information, which is communicated using structured signals.



Information/data is always physically encoded as some pattern. Creating (or breaking) patterns requires energy. Physicists have provided experimental proof of Szilárd's law about equivalence of information and energy [34], [35]. Changes in energy are measured as entropy and entropy has been more and more used in very different studies - studies of business systems [36] or football games [37]. Entropy is a proper measure when we consider the current flood of data (information?) - massive Information processing inevitably also increases entropy of our environment.

#### 4. What is Entropy ?

Status of the concept Entropy (thermodynamical or Shannon entropy - several other types of entropy have been introduced and studied) is even more messy than this of information, and also from the very beginning, from Wiener and Shannon, who actually give to this concept diametrically opposite explanations/definitions.

For Wiener entropy was opposite to information:

"Just as the amount of information in a system is a measure of its degree of organisation, so the entropy of a system is a measure of its degree of disorganisation" [21], p. 18; " It will be seen that the processes which lose information are ... closely analogous to the processes which gain entropy"; "The quantity that we here define as amount of information is the negative of the quantity usually defined as entropy in similar situations" [21], p.76.

Shannon used the word 'entropy' in opposite sense.

"The quantity which uniquely meets the natural requirements that one sets up for 'information' turns out to be exactly that which is known in thermodynamics as entropy" [38], p. 103.

This difference in views and definitions between Wiener and Shannon comes from their different research topics. Shannon considered communication and signals in communication channel without any meaning, i.e. pure data, but Wiener considered the use of these signals in control - for Wiener these signals had meaning, were knowledge. Shannon entropy of signals/data is *potential information* - if the receiver can use it for achieving its goals (proving its goal predicate true) then it becomes for receiver information.

Shannon entropy is a measure of uncertainty of signal/data which is considered as an unknown random variable  $x$  with possible values  $x_1, \dots, x_n$ , which appear with probabilities  $p(x_1), \dots, p(x_n)$ . This measure should satisfy some natural properties:

- it should be positive (this is the information what we get when it becomes available);
- it is bigger (there is more uncertainty), if probabilities  $p(x_1), \dots, p(x_n)$  are more equal (distribution is broad, more possibilities); it is smaller, if there are one or some sharp peaks (some values are far more probable);
- is additive for two independent variables.

Shannon proved, that these properties are satisfied if uncertainty (entropy) is described with function

$$H(x) = -K \sum_i p(x_i) \ln(p(x_i))$$

The value of this function does not depend at all on possible values  $x_1, \dots, x_n$  of the variable, only on their probabilities. Changing the base of the logarithm function changes only multiplicative constant  $K$ , thus usually the base of the logarithm is 2.

Examples.

1. Entropy of a single Boolean variable  $x$ , which gets both values 0,1 with equal probability:

$$H(x) = -K 2 \left( \frac{1}{2} \ln \left( \frac{1}{2} \right) \right) = K$$

By a common agreement a single Boolean variable contains just 1 bit of entropy/information, thus constant  $K$  may be set to 1 [39]. Since  $p(x) = p(\neg x)$  ( $\neg$  - negation), also  $H(\neg x) = 1$ .

2. If  $x, y$  are two independent Boolean random variables then the information, what we receive if we get the value of the function  $xy, x \vee y, x \rightarrow y$  or from any function obtained from these functions with negations of some variables, is the same, e.g.

$$H(x \vee y) = -\left( \frac{3}{4} \ln \left( \frac{3}{4} \right) + \frac{1}{4} \ln \left( \frac{1}{4} \right) \right) = -\left( \frac{3}{4} \ln 3 - 2 \right) = 0.811$$

but

$$H(x \oplus y) = 2, \quad H(x \sim y) = 2 \quad (\oplus - \text{exclusive-OR}, \sim - \text{equivalence})$$

This means, that information received from knowing, that two Boolean variables are equivalent is exactly the same what we receive from the opposite (they are antipodes) and the same what we get from receiving values of variables  $x, y$  directly.

2. If  $x, y, z$  are three independent Boolean random variables then the entropy of the value of the function

$$f(x, y, z) = (x \vee y) \neg z$$

then

$$H(f) = -\left( \frac{3}{8} \ln \frac{3}{8} + \frac{5}{8} \ln \frac{5}{8} \right) = 0.954$$

The above function could be presented also as an implication

$$f(x, y, z) = z \rightarrow (x \vee y)$$

If goal is predicate *vitamins* and IPS has already some information, e.g..

$$\text{vitamins} \Leftrightarrow \exists x(\text{vitamins}(x))$$

$$\text{vitamins}(x) \Leftrightarrow (\text{apple}(x) \vee \text{pear}(x)) \neg \text{potato}(x)$$

and IPS gets from its environment grounded information

$$a \wedge \text{apple}(a)$$

then it can deduce goal *vitamins* and perform its goal action - *eat(a)*.

## 5. Information Processing Systems

The structure of all Information Processing Systems is similar to chemoton - abstract model for the fundamental unit of life [40].

All IPS have some enclosing system, which keeps everything together and separates it from the surrounding environment. For governments this is the power structures - police, army, for cells - the cell membrane, languages are kept going by information stored in memory of language users etc.

Inside this encapsulating membrane is performed self-sustaining Information Processing with two important subsystems:

metabolism - using energy for keeping the system alive;

self-replication, which constantly revamp the system, so that it better corresponds to changes in environment (governments - elections, cells - division, language - new words invented by language users).

Information Processing is carried on in some hierarchical structure. The lowest level is grounded on external environment, the elementary items of information are patterns observed in the context. The higher levels use results of lower levels the same way - they follow patterns evolving on lower levels.

All Information Processing Systems are constantly modified and developed, levels become more complex, new levels are added or removed.

Execution of a computer program proceeds on several levels; on each level processing is grounded on results obtained in previous level:

- reading in the file of characters (obtaining information grounded by environment);
- recognizing tokens/lexems, using results from the previous level;
- recognizing the lexical structure of declarations/statements, using the results from previous level;
- the lexical/grammatical structure is used for constructing AST (Abstract Syntax Tree)
- AST is used to generate code on some low-level language (nowadays not directly in machine/processor code, but code of some virtual machine)
- the code is executed and result is used in some external device (e.g. Google automatic car) or as input to some other program, i.e. output from this program becomes the ground for the next program.

Rapid development of Information Processing and Internet have made Programming Languages a very dynamic IPS; new languages appear constantly - Dart, Go, Ceylon, Livescript, Elixir, F#, M#, Opa, ... .

Like all Information processing Systems programming languages also evolve under pressures external forces - simplicity of use (compare e.g. Cobol, Fortran with Ruby/Python), functional needs (the C language family versus Javascript)

Development of living beings, e.g. humans (the highest, most complex structure of IP) also proceeds step-by-step, every step introduces a higher-level Information Processing structures:

- A child in his first, sensory motor stage of development (0..2 years), when child knows the world only through movements and sensations acquires (grounded) knowledge only through sensory experiences, when manipulating objects, e.g. sucking his/her toys - is this eatable or not? This is the first level of child's cognitive development described by psychologist Jean Piaget [41]. Around 7 to 9 months of age begins memory development, e.g. child understands, that objects do not disappear when they are removed.



- On the next, preoperational stage (years 2..7) appears symbolic play - child uses impressions and images from his memory to start make-believe games and fantasy roles - dressing up as their favourite super hero or as the “mommy” or “daddy”. Here occurs also very intensive development of language - "vocabulary burst" [42]. All these activities are based (grounded) on created in child's head concepts and connections/relations between these concepts.
- On the next, concrete operational stage (age from around seven until eleven years) appears better understanding of mental operations (i.e. that operations can be reversible), understanding of induction - from a specific experiences may be derived general principles; these abilities allow to understand logically and inductively derived concepts, e.g. natural numbers.



**Figure 1.** Logical grounding of concept *fruit*: apple and pear are fruits, but not potato



**Figure 2.** Inductive grounding of a concept: "3" is what is common on these pictures

- Child also becomes aware that he is a member of society and that others have their own understanding of the world, which may be different and provide new information; understanding that others provide useful information creates friends and gangs, i.e. there appear new IPS on new, higher level.
- On the last, formal operational stage which begins approximately at age twelve people develop ability of handling abstract concepts, logical thought, deductive reasoning and systematic planning.

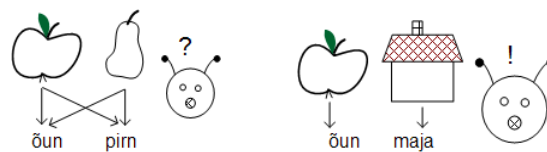
Piaget did not consider children's intellectual development as a quantitative process of storing more and more information in their memory, but as a process of qualitative changes in their mental machinery; Children develop new levels of Information Processing as they gradually process through these four stages.

And there may be more stages. When Piaget followed his children it was believed, that human brain becomes mature somewhere in the middle of 20s [43] and after that begins decay, decrease of brain cells. But recent research shows, that (possibly under pressure of changes in life style - pursuing post-secondary education, starting a career, independence and developing new social and family relationships) brain development continues, new brain cells appear and wiring of brain connects different regions to facilitate cognitive abilities [44], [45].

Humans invented for collecting information, for developing more complex structures, functions and understanding of their environment, thus reducing their entropy a totally new tool – language. Language moved the process of collecting information about their environment needed for survival from the level of individuals to the level of the whole Mankind. Language is the Mankind's IPS for modelling the World. Language reflects structure of the World and thus helps mankind to survive,

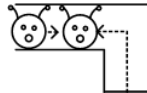
develop and advance and (in the limit) converges to a similar structure, i.e. has entropy close to the entropy of the World which it describes.

Emergence of common understanding of signals, i.e. emergence of words has been studied and modelled in agent communities. Emergence of common vocabulary in a society of agents is based (grounded) on disambiguation of meanings when objects are presented in different contexts [46], [47], [48]. Made in simulations measurements of the entropy of language show, that language continues to develop also when agents already well understood each other; entropy of language steadily increases, but remains smaller than the entropy of the environment which language models [46].



**Figure 3.** Disambiguation (grounding of word meaning on context). When agent meets in an environment of two objects and the other agent says two words: "oun", "pim" then certainly there is no way to understand, what is what. But if he then sees one of these objects (apple) in different context and again hears the word "oun" then he can place all words for corresponding objects: "oun" = apple, "maja" = house, "pim" = pear" [46]

Disambiguation of meaning can be based also on local spatial context [46].



**Figure 4.** Disambiguation using local spatial context. When agent 1 (on left) first meets agent 2 (right) who says "tupik", agent 1 can not infer anything. But when he continues and finds, that the passage is dead end, he can infer from the local spatial context, that "tupik" = "dead end" [46]

After obtaining first common denotations (words) for external objects, further development of language can use already established semantics.

All business organizations are Information processing Systems which operate with huge amounts of data. They produce it every time a product or ingredient is received, produced, shipped or sold and for all these operations are traditionally separate sub-units - Procurement, Production, Research, Sales, Advertising and Public relations etc departments. based on hierarchical multi-level structure of Information Processing, where processes on higher level use results, i.e. are grounded on results of processes on lower level [49],[50].

## 6. Conclusions

Here was proposed an unified view of Information Processing in different natural Information Processing Systems - living cells and living organisms, language, business organisations, governments etc.

All Information Processing systems (IPS) are finite and have *goals*, their purpose is to perform some *actions* aimed to fulfil their goals. They survive only by constantly processing information about threats and opportunities in the world around them. Goals of living systems are metabolism (getting energy needed for their functioning), and reproduction, goals of businesses - produce some goods and/or services etc.

To achieve their goals, IPS use information they receive from their environment and they actively search it (child puts everything in mouth: "Is it eatable?").

Data from external environment is encoded as certain patterns - visual, chemical, mechanical, olfactory etc, i.e. It is grounded - does not have meaning for the environment itself, but can be mapped to stored patterns by the IPS; all IPS are pattern-matching systems. The IPS have to learn meaning of environment patterns. Survival of IPS is based on information they receive from their environment, thus in the first stages of development they try to keep everything for themselves (child in kindergarten: "This is MY puppy!"). But when they develop and receive more than needed for survival, they start communicate, share information and things with other IPS belonging to the same level of development ("You can play with my puppy!"). Information sharing makes them more effective, speeds up their information-gathering and results with new level of more complex IPS – friends, gangs, political parties, states etc. These next-level IPS behave the same way: first they try to establish some borders, areas of influence etc, but then start cooperation with similar IPS - gangs find similar gangs in other countries, countries join EU etc. The next-level IPS are always more complex, use more complex information (personal perceptions are replaced with language communication, first face-to-face, then from communication networks, then Internet) and more effective in reducing their inner entropy and in the same time increasing entropy of their environment.

Goals of IPS can be understood as predicates - their *goal predicates*, the input data - as object variables and logical connectives, defining predicates. Information Processing in an IPS is logical derivation (computation) of goal predicates from inputs and this increases entropy in the environment; this increase of entropy is subject to the Maximum Entropy Production Principle [51].

## References

- [1] Konrad Zuse, Rechner Raum, Elektronische Datenverarbeitung, vol. 8, pages 336-344, 1967
- [2] Konrad Zuse, Rechner Raum, Friedrich Vieweg & Sohn, Braunschweig, 1969.
- [3] Seth Lloyd Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos. Vintage 256 pp. 2007
- [4] Jérémie Cabessa & Hava T. Siegelmann, The Computational Power of Interactive Recurrent Neural Networks, Neural Computation, 2012
- [5] Burgin, M. Super-recursive algorithms. Monographs in computer science, Springer 2005., ISBN 0-387-95569-0
- [6] T. Stonier. Information and the Internal Structure of the Universe. Springer Verlag 1990
- [7] Shannon C. E., Weaver W. The Mathematical Theory of Communication. Urbana University of Illinois Press, 1949.
- [8] Whitworth, B. The emergence of the physical world from information processing. Quantum Biosystems 2010, 2(1), pp. 221-249
- [9] The Digital Universe in 2020. <http://www.emc.com/leadership/digital-universe/index.htm>
- [10] <http://www.worldometers.info/world-population/#pastfuture>
- [11] <http://www.worldometers.info/world-population/#pastfuture>
- [12] Big Data, for better or worse: 90% of world's data generated over last two years. <http://www.sciencedaily.com/releases/2013/05/130522085217.htm>
- [13] Mark Andreessen. Why Software Is Eating The World. <http://online.wsj.com/news/articles/SB10001424053111903480904576512250915629460>
- [14] <http://www.oxforddictionaries.com/definition/english/set?q=set>
- [15] <http://www.oxforddictionaries.com/us/words/is-a-penguin-a-bird>
- [16] [http://gamasutra.com/blogs/NilsPettersson/20130116/184876/What\\_is\\_a\\_game\\_and\\_why\\_do\\_we\\_play.php](http://gamasutra.com/blogs/NilsPettersson/20130116/184876/What_is_a_game_and_why_do_we_play.php)
- [17] Burgin, M. Theory of Information. world Scientific 2010, ISBN-13 978-981-283-548-2

- [18] - T.L. Duncan, J.S. Semura. Information Loss as a Foundational Principle for the Second Law of Thermodynamics. *Foundations of Physics* (2007) 37: 1767-1773
- [19] - T. G. Spiro, W. M. Stigliani. *Environmental Issues in Chemical Perspective*. Suny Press 1980
- [20] - Brain-Like Chip May Solve Computers' Big Problem: Energy. *Discover Magazine*, Oct 2009, online: <http://discovermagazine.com/2009/oct/06-brain-like-chip-may-solve-computers-big-problem-energy>
- [21] - Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine*. Paris, (Hermann & Cie) & Camb. Mass. (MIT Press), 1948
- [22] - G. G. Scarrott. The Nature of Information. *Computer Journal* 32:3, 1986, pp. 262-266
- [23] - Williams, M.A. Representation = Grounded Information. *PRICAI 2008: Trends in Artificial Intelligence. Lecture Notes in Computer Science Volume 5351*, 2008, pp 473-484
- [24] - Shannon, C.E. The mathematical theory of Communication. *Bell System Technical Journal*, v.27 No. 1, pp 379-423; No. 3, pp. 623-656
- [25] - Shannon, C.E. *Collected Papers*. 968 p., Wiley-IEEE press, 1993, ISBN-13: 978-0780304345
- [26] - <http://www.eoht.info/page/Shannon+bandwagon>
- [27] - Wiener, N. *Cybernetics, or Control and Communication in the Animal and the Machine*. Cambridge, MIT Press 1948; online: <http://www.scribd.com/doc/66686625/Wiener-Norbert-Cybernetics-Or-Control-and-Communication-in-the-Animal-and-the-Machine>
- [28] - Rolf Landauer: "Irreversibility and heat generation in the computing process," *IBM Journal of Research and Development*, vol. 5, pp. 183–191, 1961.
- [29] - Reeb, D., Wolf, M.M.. (Im-)Proving Landauer's Principle. arXiv:1306.4352 [quant-ph]
- [30] - Schiller, D. (1996). *Theorizing communication*. New York: Oxford University, p. 168
- [31] - Roger C. Schank. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*. 256pp, Psychology Press, 1977 ISBN-10: 0898591384, ISBN-13: 978-0898591385
- [32] - M. Burgin. *Theory of Information. Fundamentality, Diversity and Unification*. World Scientific Publishing Co, 2010, p 99
- [33] - K. Hefner (Ed.). *Evolution of Information Processing Systems*. Springer-Verlag 1992
- [34] - Machta, J. Entropy, information, and computation. *Am. J. Phys.* 67 ~121, December 1999
- [35] - S. Toyabe, T. Sagawa, M. Ueda, E. Muneyuki, M. Sano (2010) Experimental demonstration of information-to-energy conversion and validation of the generalized Jarzynski equality. *Nature Physics* 6, 988–992
- [36] - Klimenko A.Y. Entropy and Equilibria in Competitive Systems. *Entropy* 2014, 16, 1-22
- [37] - Couceiro M.S., Clemente F.M., Martins F.M., Machado J.A. Dynamical Stability and Predictability of Football Players: The Study of One Match. *Entropy* 2014, 16, 645-674
- [38] - Claude E. Shannon, Warren Weaver. *The Mathematical Theory of Communication*. Univ of Illinois Press, 1949. ISBN 0-252-72548-4
- [39] - Jaynes E.T.. *Information Theory and Statistical Mechanics*. *Phys. Rev.* 106, 620–630 (1957)
- [40] - T. Gánti: *The Principles of Life*, Oxford University Press (2003) ISBN 9780198507260
- [41] - Piaget, J., Gruber, H.E.; Voneche, J.J. eds. *The essential Piaget*. New York: Basic Books, 1977
- [42] - McCarthy, D. Language development in children. In L. Carmichael (Ed.), *Manual of child development* (pp. 492–630). New York: Wiley 1954
- [43] - Pujol J, Vendrell P, Junqué C, Martí-Vilalta JL, Capdevila A. When does human brain development end? Evidence of corpus callosum growth up to adulthood. *Ann Neurol.* 1993 Jul;34(1):71-5.
- [44] - Lebel, C., Beaulieu, C. Longitudinal Development of Human Brain Wiring Continues from Childhood into Adulthood. *The Journal of Neuroscience*, 27 July 2011, 31(30): 10937-10947
- [45] - Ramscar M., Hendrix P., Shaoul C., Milin P., Baayen H. The Myth of Cognitive Decline: Non-Linear Dynamics of Lifelong Learning. *Topics in Cognitive Science Volume 6, Issue 1, pages 5–42, January 2014*
- [46] - J. Henno. *Emergence of Language: Hidden States and Local Encironments*. *Information Modelling and Knowledge Bases XIX*, Hannu Jaakkola, Yasushi Kiyoki & Takahiro Tokuda (eds), IOS Press Amsterdam-Berlin-Oxford-Tokyo-Washington DC, ISBN 978-1-58603-812-0, pp 170-181
- [47] - L. Steels (2006) How to do Experiments in Artificial Language Evolution and Why. In Cangelosi, A., Smith A. and Smith K., editor, *Proceedings of the 6th International Conference on The Evolution of Language (EVOLANG6)*, London
- [48] - L. Steels, P. Vogt (1997). Grounding adaptive language games in robotic agents. In C. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life*, Cambridge MA and London, 1997. The MIT Press
- [49] - Galbraith, J.R. *Designing Complex Organizations*. Reading. MA: Addison-Wesley, 1973.
- [50] - Anderson, C.. What are the Top Ten Core business processes?, *Bizmanualz*, July 22nd, 2009.
- [51] - Dewar R. Information theory explanation of the fluctuation theorem, maximum entropy production and self-organized criticality in non-equilibrium stationary states. *J Phys A* 36: 631, 2003

# An Explorative Cultural-Image Analyzer for Detection, Visualization, and Comparison of Historical-Color Trends

Yoshiko ITABASHI <sup>a,1</sup>, Shiori SASAKI <sup>b</sup> and Yasushi KIYOKI <sup>c</sup>

<sup>a</sup> *Keio Research Institute at SFC Keio University*

<sup>b</sup> *School of Media and Governance Keio University*

<sup>c</sup> *Faculty of Environment and Information Studies Keio University*

**Abstract.** This paper presents an explorative cultural-image analyzer and its application for comparative analyses of cultural arts and crafts. The goal of this system is to provide a new image-exploration environment that reflects the diversity of humans' sense of color and the breadth of cultural human knowledge by detecting and visualizing characteristic historical color-trends within cultural-image data sets. The primary components of this system are the two explorative analysis methods with feature estimation and evaluation of culture-dependent colors: (a) image-group exploration and (b) color exploration. The system visualizes the distinct differences among image groups aggregated by the attributes such as author, era, region, etc. and provides notable images for users through the image-group exploration method. In addition, the system visualizes the subtle differences of colors in images and provides a key to analyze cultural art works by the color-exploration method, with a zooming function for color distributions and cultural-color name estimation. By utilizing the existing annotations and attributes that are available for most images, the system analyzes the differences of colors among image-groups defined by statistical analysis and visualizes the representations of each image-group on an overview map. This system enables a user to analyze the characteristics of a collection of cultural art works by browsing the representative images of each image-group, exploring the specified culture-dependent colors with high accuracy, and observing subtle differences of colors among image-groups according to culture-dependent color names at a glance.

**Keywords.** Multimedia, Image processing, Color naming, Visual analytics, Cross-cultural computing.

## 1. Introduction

To analyze cultural arts and crafts, it is important to compare several media groups and discover common features and subtle differences between them [12][13][17]. There are a lot of contexts to create image groups of cultural arts and crafts for analyses. For example, we are able to create groups based on attributes of works such as author, media type, school, style, time/age, area/region, etc.. This paper presents an explorative cultural-image analyzer and its application for comparative analyses of cultural arts and crafts. The system visualizes the distinct differences among image groups aggregated

---

<sup>1</sup> Corresponding Author: Yoshiko Itabashi, Keio Research Institute at SFC Keio University, Endo 5322, Fujisawa, Kanagawa, Japan; E-mail: itabasiy@sfc.keio.ac.jp



by the attributes such as author, school, era, region, etc. and provides notable images for users through the image-group exploration method. The system reflects the diversity of humans' sense of color and the breadth of cultural human knowledge by detecting and visualizing characteristic historical color-trends within cultural-image data sets.

The objective of this research is to realize a new analyzer for computing differences in cultural studies fields for discovering new knowledge based on actual aspects and contexts. Based on the approach of Differential Computing [18], this system is realized to analyze cultural and artistic activities of human being by automatically detecting and visualizing historical trends of colors in arts and crafts over time.

In this system, the images, their metadata and color distributions are analyzed, the intuitive overview of images is displayed for users to make it easier to understand. The main feature of this system is that the system provides the methods of color exploration with cultural-color knowledge to refine color analysis. Using the method, the users of this system are able to explore the relation between metadata of image and color, and obtain the new insight about the subtle differences of colors.

Generally, it is difficult for people to grasp an overview of many images. People can only recognize the features of a few images from their thumbnails at a glance. Even using the MDS (multi-dimensional scaling) technique [2][36][37], a display cannot provide clear information because many thumbnails are simultaneously displayed on a screen of limited size. The number of dimensions of image features is beyond the human's visual perception and the abilities of simple displays.

In the fields of image processing and CBIR (content-based image retrieval) [24] [6] [7], color is considered as one of the most important features. The color information of each image is extracted as a color histogram and vectorized by a reference color set. The reference colors are defined by colors of the color histogram bins. The number of the color histogram bins should be approximately 100 for efficient processing [35]. However, the degree of color variety differs among cultures, and each color name is designated uniquely in a language.

Berlin and Kay examined basic color names from the ethnobiology perspective [3]; they defined a set of color names using common vocabularies, such as "red", "blue", "black", and "white", although the range of each color name differs by culture. E. Rosch defined these basic color names as prototypes [11] from the cognitive psychology perspective. Additionally, local color names, such as collections in DIC Color Guide [9], including both local and traditional color names, have been used by people living in specific areas/cultures. Generally, the number of culture-dependent color names is greater than 100. In this sense, the processing of existing CBIR technique decreases the diversity of humans' sense of color.

There are vast amounts of images with annotation data archived in digital archives, electronic libraries and on-line museums on the WWW. Many collections of traditional arts and crafts owned by museums around the world exist in wide-area networks with retrieval services using annotation data. Furthermore, automatic annotation from image contents has been widely studied in recent years [8].

Based on these backgrounds, this paper presents an explorative image analysis system that visualizes the relations between color distributions and annotations of images. The visualized results enable users to explore characteristic images and gain insight about the distinct colors used in a collection of images. The question "compared to what?" lies at the heart of various information visualization and data mining

techniques [14] [33]. The essence of our system is the ability to compare color features among image-groups grouped by annotations and culture-dependent color features.

The process of our system for analyzing target image groups consists of the following steps. First, the system computes the color histogram of all images using approximately 100 bins, compares the image groups with each other and extracts distinctive features using statistical analysis. Second, the system outputs two feature representations of each image group using aggregation techniques. One technique uses a color histogram to visualize distinctive color features of the image group, and the other uses a representative image of the image group generated by a similarity calculation. Third, the system displays an overview map of the representations of image groups in a style suitable for the attributes of image groups. Fourth, the system presents the thumbnails sorted by representative features and linked to the details of the images.

For color information, the system starts the analysis with an overview of the image groups. First, to zoom in, the system rescans the target images to compute color distributions of the neighboring colors based on the colors that a user selects for the analysis. Second, the neighborhood distance and granularity of color quantization are optimized for the specified colors and purpose. Third, the system displays the recomputed color distribution and estimates a set of color names from culture-dependent color names [12]. Finally, the system aggregates and visualizes culture-dependent colors and the similarity between each image group. Zoom and filter processes [27] are applied to both image-group computing and color-distribution computing. Therefore, the system enables high-level explorative analysis of color features depending on image attributes such as author, country, and era. The representation of color by culture-dependent color names enables people of various cultural backgrounds to easily understand the characteristic of colors of image groups.

## 2. Basic Method and System Architecture

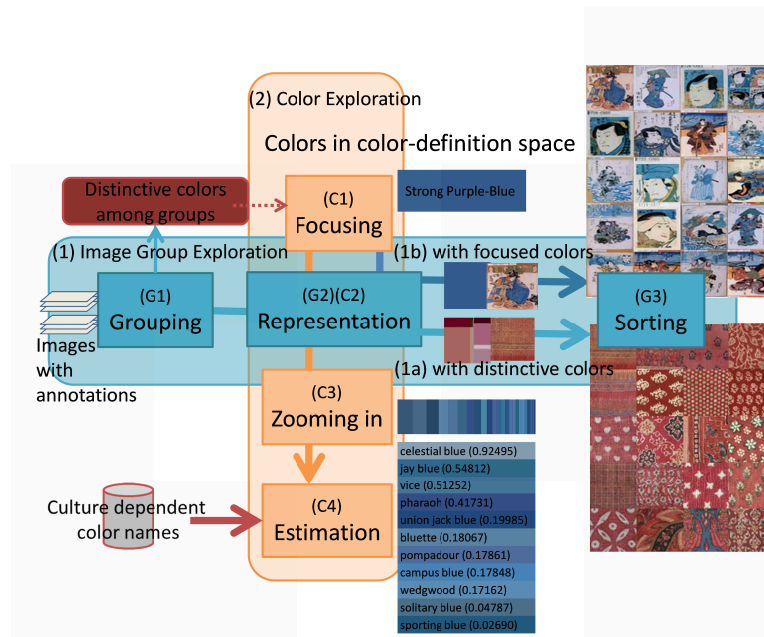
Our system has two exploration methods: (a) image-group exploration and (b) color exploration. Figure 1 shows the system structure.

The sky-blue boxes and arrows in Figure 1 indicate the image-group exploration process. The orange boxes and arrows in Figure 1 denote the color-exploration process.

The image-group exploration consists of the following three functions. The first function is grouping and distinctive color extraction (Figure 1: G1). The system defines image groups according to a viewpoint selected by the user and extracts distinctive colors via a statistical analysis. The viewpoint is selected from text annotations such as author, city and era and numerical annotations such as spatiotemporal parameters. The second function is computing representations of image groups (Figure 1: G2). The system creates two types of representations of each image group based on the colors focused on by the user. Two examples of representations are shown in the right side of Figure 1. The upper images are the results based on ‘Strong Purple-Blue’, and the lower images are the results of distinctive colors. The representations are mapped on an overview map that is suitable for visualizing image groups (Figure 2). The third function is sorting images of each image group with representative features (Figure 1: G3). The user can access the details of the image by selecting a thumbnail.

The color exploration consists of the following three functions. The first function is focusing colors (Figure 1: C1). The system calculates the candidate colors that

should be focused on by users using a statistical technique. The second function is zooming in on the color distributions of images (Figure 1: C3). The system rescans the target images to calculate the color distribution of the neighboring colors of the specified color. The neighborhood distance and the granularity are optimized for specifying colors. The third function is estimation of culture-dependent color names (Figure 1: C4). This process transforms the color-distribution data to human-readable information such as color names.



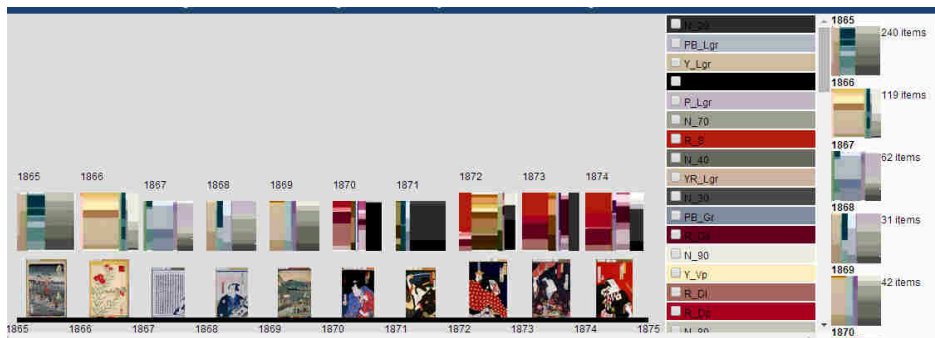
**Figure 1.** Structure of the image-exploration system: (1) (1a) image-group exploration with distinctive color, (1b) image-group exploration with focused color, and (2) color exploration

The details of the image-group exploration method are described in Section 3, and the details of the color exploration method are described in Section 4. Both are discussed in detail using examples from experiments. The target data of the experiments are 8744 images of Ukiyoe from the Tokyo Metropolitan Library [32].

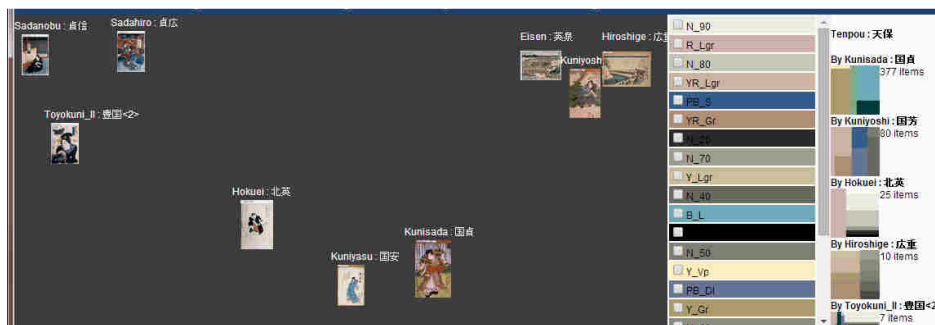
As an experimental environment, we selected 130 colors from the Color Image Scale (CIS) [19] as reference color set for the color histograms used for image-group exploration. Hereafter, the colors used for the color-definition space are described as Red, Yellow-Red, etc. according to the colors in CIS. DIC Color Guide [9] and some color collections [23] are used as culture-dependent colors. We calculate the color distance with the CIE2000 formula [20] or a corn model of HSV using a value weight of 2.0. The image color histograms are generated using the color-definition of CIS and calculation of the color distance with a corn model of HSV.



(2a)



(2b)



(2c)

**Figure 2.** Examples of overview maps: (2a) geographical map of the characteristics of Chintz/Sarasa grouped by country of production (18th century) [21][29][30][31][34][10], (2b) timeline map of the characteristics of Ukiyoe grouped by year (1865-1874) [32], (2c) the characteristics of Ukiyoe grouped by author in the Tenpou era using MDS (multi-dimensional scaling) [32][2][5]

### 3. Image-Group Exploration Method

#### 3.1. Grouping and distinctive color extraction

Generally, most images are archived with annotation data. In our system, image groups are defined by the annotation data. Text annotation such as author, city and era are major grouping keys. Numerical annotations such as the spatiotemporal parameters are also useful for grouping images. Users can define the image groups as their viewpoints for the analysis.

Target images are vectorized as color histogram with color space quantization over approximately 100 dimensions. The quantization number meets the general requirement of image retrieval [35]. This quantized color space is called the “color-definition space” in our system.

The process to extract distinctive colors among groups consists of the following steps. (1) Calculating the average color distribution of all selected images and each image group. (2) Extraction of distinctive colors from the colors of the color-definition space using ANOVA (analysis of variance) for a single factor.

For the purpose of statistic testing of differences in the distributions of colors by group, we apply the analysis of variance (ANOVA) method to each color of the color-definition space. ANOVA is a hypothesis testing method to test the equality of means. The null hypothesis is that the difference between groups’ mean values is significant. Distinctive colors are the colors of the null hypothesis. We reject this hypothesis if the p-value (possibility) is 0.01 or less.

#### 3.2. Focusing colors from the color-definition space

Users of our system can specify colors for analysis from the color-definition space that includes all colors. We assume that the set of significant colors identified by variance analysis is useful for representation of the differences among image groups and define this set as the “distinctive colors”.

#### 3.3. Representation of image groups

We provide two aggregation methods for each image group with the selected colors. One is an averaged color feature that is calculated as the mean of the color distribution for each image group. The other is a differential color feature that is calculated as the difference between the group mean of a color distribution and the grand mean of the color distribution.

Aggregation methods are applied to image groups on subspaces contracted by the focused colors. Our system has the following three aggregation functions.

- (1) A function to calculate average color features from all the colors used in an image group
- (2) A function to calculate the differential color features on distinctive colors
- (3) A function to calculate the average color features on focused colors

(1) is used for extracting the average color features from an image group. (2) is used for extracting a set of characteristic colors of an image group compared to the other image groups. (3) is used for extracting the difference of averaged combination of focused colors between image groups.

The system outputs two representations of each image group's characteristic using aggregation techniques. One representation is a color histogram to visualize aggregated color features of the image group, and the other is a representative image of the image group that is determined by a similarity calculation. The similarity is calculated using the inner product with the proper normalization. The representative image is more impressive than the color histogram, but the visualized color histogram can give more explicit information.

### 3.4. The overview map of image groups and sorting

Users of our system can select a suitable style for the overview that depends on the group definition. If the group definition has a location attribute, a geographical map is suitable. If the group definition has a time attribute, a timeline map or timeline bar is suitable. If the user wants to display the relations of the features among image groups, an MDS map visualizes the similarity of features among image groups.

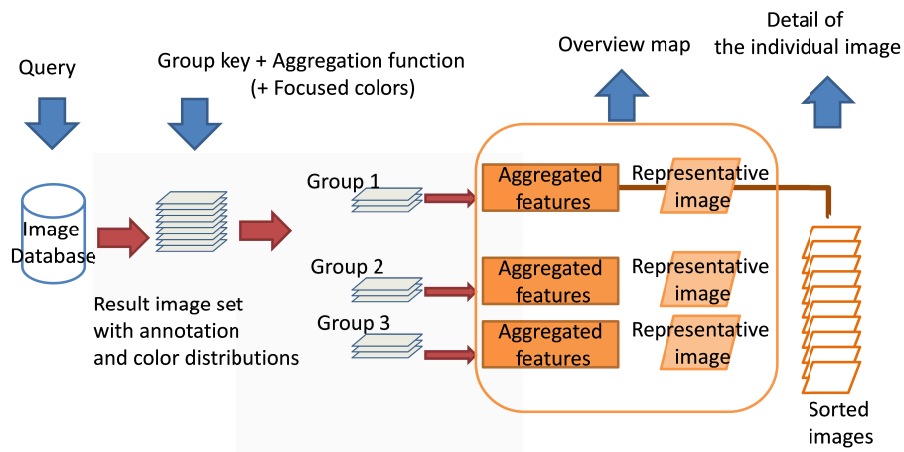


Figure 3. User interface flow chart of the image-exploration system.

Figure 3 shows the relation between overview map and sorting. “Query” and “Group key+Aggregation function (+ Focused colors)” in the left part of the figure are input parameters. The system shows the overview map to the user. The user is able to select one image and access the information of the image, after grasping the overview of groups. The system presents the thumbnails sorted by aggregated features and the links to the details of the images. The calculation of similarity is executed for each operation. Thus the calculation is computationally expensive. Our system calculates the similarity by inner-product or cosine.

### 3.5. Experiments on Image-group exploration

In this section, we present several examples of the application using our method to show the applicability of our system for images of fine art. The target data of the experiments are 8744 images of Ukiyoe from the Tokyo Metropolitan Library [32].

Each image of this collection has the annotation data, including the author name and the date of publication. Search results of 1334 images can be obtained if you search for Ukiyoe published in 10 years around the Meiji Restoration (1868-1869), when the modern Meiji era started in Japan, before and after. These images are grouped by the annotation data and these groups are compared each other using color distribution. This section shows the examples of aggregated features and representative images. And one example of overview map is shown.

Table 1 shows the aggregation results. We can see that the target images were divided into two groups of the publishing year before and after 1869. The second column indicates the average color features on full color. These features are similar because of common style of Ukiyoe. The third column indicates the differential color features on distinctive colors. These features are different by the effect of the calculation of difference and the reduction of color number. The fourth column shows the distinctive color list. One of most significant color in distinctive colors is Strong Red in CIS. The color was used more widely after the Meiji Restoration.

When the 1334 images are grouped by author, we can see that there are 16 authors who published more than three pieces of Ukiyoe. The system calculates the 71 distinctive colors among authors. Strong Red is one of distinctive colors, but the color is less significant than Dark Red. This indicates that the popularity of Strong Red after the Meiji Restoration may be independent of the author.

**Table 1.** Examples of representations of images of Ukiyoe of 5 years published around 1869 (retrieval results of Ukiyoe published from 1864 to 1873).

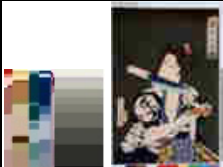
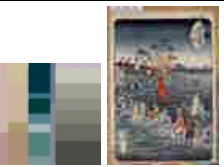







	The average color features on full color	The differential color features on distinctive colors	The list of distinctive colors (58 colors)																																																												
<b>Before 1869</b> 725 images			<table border="1"> <tr><td>R_S</td><td>YR_Dk</td><td>P_Dl</td></tr> <tr><td>N_20</td><td>Y_Lgr</td><td>B_Gr</td></tr> <tr><td></td><td>BG_Dgr</td><td>B_S</td></tr> <tr><td>R_Dk</td><td>R_Gr</td><td>RP_S</td></tr> <tr><td>N_30</td><td>PB_Gr</td><td>RP_Vp</td></tr> <tr><td>R_Dl</td><td>YR_L</td><td>GY_Dgr</td></tr> <tr><td>N_40</td><td>Y_Dl</td><td>P_S</td></tr> <tr><td>YR_Lgr</td><td>P_Lgr</td><td>G_Gr</td></tr> <tr><td>R_Dgr</td><td>B_Dl</td><td>RP_Dgr</td></tr> <tr><td>YR_Gr</td><td>RP_Dl</td><td>YR_P</td></tr> <tr><td>PB_Dgr</td><td>P_Dgr</td><td>Y_Dk</td></tr> <tr><td>N_70</td><td>B_Lgr</td><td>YR_S</td></tr> <tr><td>R_Dp</td><td>PB_Dl</td><td>GY_Dk</td></tr> <tr><td>N_80</td><td>R_L</td><td>Y_Dp</td></tr> <tr><td>B_Dgr</td><td>Y_Gr</td><td>Y_S</td></tr> <tr><td>R_Lgr</td><td>RP_L</td><td>R_P</td></tr> <tr><td>N_50</td><td>Y_Dgr</td><td>P_L</td></tr> <tr><td>N_60</td><td>YR_Dp</td><td>P_Vp</td></tr> <tr><td>BG_Dl</td><td>BG_Gr</td><td></td></tr> <tr><td>YR_Dgr</td><td>RP_Gr</td><td></td></tr> </table>	R_S	YR_Dk	P_Dl	N_20	Y_Lgr	B_Gr		BG_Dgr	B_S	R_Dk	R_Gr	RP_S	N_30	PB_Gr	RP_Vp	R_Dl	YR_L	GY_Dgr	N_40	Y_Dl	P_S	YR_Lgr	P_Lgr	G_Gr	R_Dgr	B_Dl	RP_Dgr	YR_Gr	RP_Dl	YR_P	PB_Dgr	P_Dgr	Y_Dk	N_70	B_Lgr	YR_S	R_Dp	PB_Dl	GY_Dk	N_80	R_L	Y_Dp	B_Dgr	Y_Gr	Y_S	R_Lgr	RP_L	R_P	N_50	Y_Dgr	P_L	N_60	YR_Dp	P_Vp	BG_Dl	BG_Gr		YR_Dgr	RP_Gr	
R_S	YR_Dk	P_Dl																																																													
N_20	Y_Lgr	B_Gr																																																													
	BG_Dgr	B_S																																																													
R_Dk	R_Gr	RP_S																																																													
N_30	PB_Gr	RP_Vp																																																													
R_Dl	YR_L	GY_Dgr																																																													
N_40	Y_Dl	P_S																																																													
YR_Lgr	P_Lgr	G_Gr																																																													
R_Dgr	B_Dl	RP_Dgr																																																													
YR_Gr	RP_Dl	YR_P																																																													
PB_Dgr	P_Dgr	Y_Dk																																																													
N_70	B_Lgr	YR_S																																																													
R_Dp	PB_Dl	GY_Dk																																																													
N_80	R_L	Y_Dp																																																													
B_Dgr	Y_Gr	Y_S																																																													
R_Lgr	RP_L	R_P																																																													
N_50	Y_Dgr	P_L																																																													
N_60	YR_Dp	P_Vp																																																													
BG_Dl	BG_Gr																																																														
YR_Dgr	RP_Gr																																																														
<b>On and after 1869</b> 609 images																																																															

Table 2 shows a part of the results of grouping by author of the Ukiyoe images published in 5 years before 1869, when the modern Meiji era started in Japan, and Table 3 shows the results of grouping by author of the 609 Ukiyoe images published in 5 years on and after 1869. The average color features on full color are shown in 4th column and the differential color features in distinctive colors are shown in 5th column for both tables. The Ukiyoe before 1869 are created by 12 authors and these on and after 1869 are created by 7 authors. The color differences between authors are represented clearly, especially for the Ukiyoe before 1869. As observed, the number of colors is reduced, and the features of the group are visualized clearly such that even

users who are not familiar with color can understand the distinctive colors of each collection easily.

The system calculates the 67 distinctive colors among authors for created before 1869 and the 73 colors for created ones on and after 1869. Red colors are not included in the distinctive colors for before 1869, and the distinctive colors for on and after 1869 contain four red colors (Dark Red, Strong Red, Deep Red, and Vivid Red in CIS). The differences of red color area among the authors are not much significant. This indicates that the differences of red colors in Table 1 don't depend on the author.

**Table 2.** A part of representations of images of Ukiyoe grouped by author (retrieval results of Ukiyoe published from 1864 to 1868).

Author	Author in Japanese	Number of images	The average color features on full color		The differential color features on 67 distinctive colors	
Kunichika	国周	275				
Kunisada	国貞	182				
Kunisada_II	国貞<2>	82				
Yoshiiku	芳幾	45				
Hiroshige_II	広重<2>	33				
Kuniteru	国輝	31				
Yoshitoshi	芳年	18				



**Table 3.** Examples of representations of images of Ukiyoe grouped by author (retrieval results of Ukiyoe published from 1869 to 1873).

Author	Author in Japanese	Number of images	The average color features on full color	The differential color features on 73 distinctive colors
Kunichika	国周	519		
Chikashige	周重	29		
Kunisada_III	国貞<3>	21		
Ginkou	銀光	10		
Yoshiiku	芳幾	6		
Kuniteru	国輝	6		
Yoshitora	芳虎	5		
Kunuteru_II	国輝<2>	11		

Table 4 shows the results of grouping by publishing year for the Ukiyoe images published in 4 years before around 1869. The system calculated the 35 distinctive colors that include Dark Red, Strong Red, and Deep Red. 3th column shows the average color features on full color, 4th column shows the differential color features on distinctive colors, and 5th column shows the average color features on focused colors (Dark Red, Strong Red, and Deep Red). The average color features on focused red colors visualizes clearly that red colors were used a little before 1869 and hues of red color were changed with the years.

**Table 4.** Examples of representations of images of Ukiyoe grouped by year (retrieval results of Ukiyoe published from 1867 to 1870).

Year	Number of images	The average color features on full color	The differential color features on distinctive colors	The average color features on focused colors (Dark Red, Strong Red, and Deep Red)
1867	62			
1868	31			
1869	42			
1870	91			

Figure 3 shows an example of a timeline overview map. This map visualizes the image groups of the Ukiyoe collection grouped by year of publication. Representative features are differential color features on distinctive colors. On the left side of the map, visualized color histograms and representative images of Ukiyoe are displayed in time-series (1865 – 1874). On the middle part of the interface the distinctive color list is displayed with checkboxes. On the right side of the interface, the images of selected Ukiyoe (1870) are shown in the order of similarity to the distinctive color distribution shown in the left timeline map. We can explore Ukiyoe images with the simple interface and recognize the differences at a glance.



**Figure 3.** The timeline map screenshot of the characteristics of Ukiyoe grouped by year (1865 – 1874) and sorted images of 1870.

## 4. Color Exploration Method

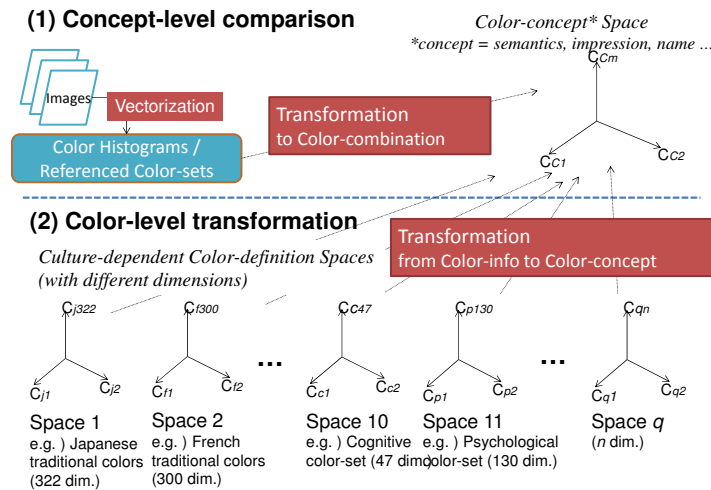
### 4.1. Focusing colors from color-definition space

Users of our system can specify colors from the color-definition space for analysis. The number of colors affects the performance of the next zooming process. Users should select less than five colors.

### 4.2. Zooming-in on a color distribution with optimization of the color set and range

To extract subtle differences in specified colors (especially distinctive colors) in multiple images, our system rescans the target images for creating a color histogram of the neighboring colors of the specified color. The neighborhood distance and granularity of color quantization are calculated based on the color distance (e.g., CIE2000). These parameters depend on the saturation of the specified color and humans' sense of color.

We have already proposed a method of treating culture-dependent colors by color space transformations and analyzing the color diversity and semantics at a concept-level [12]. The main feature of the method is characterized as a two-level comparison as shown in Figure 4: (1) Concept-level comparison and (2) Color-level space transformation. Using Color-level transformation. The color-associated terms in different color-definition spaces are mapped on the same space as the images. The images and color-associated terms are thus comparable on the concept-level.






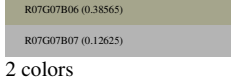
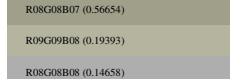
**Figure 4.** Two-level comparison of our Cross-cultural Computing method [12]

We apply this method of cross-cultural image computing [12], which allows comparing cultural differences and similarities using color information of image data in the concept level, to estimate culture-dependent color name in our system.

For the estimation of neighboring colors of a target color, we define a neighborhood subspace of color-concept space, and transform culture-dependent

colors via the vectors on the subspace. The definition of the subspace of color-concept space determines a resolution of color difference for rescanning.

**Table 5.** Examples of the neighboring colors in the color set for rescanning around ‘Strong Red’ and ‘Light Gray’ (The neighborhood distance is one sixth of the distance between pure red and pure white. The color distance is calculated with CIE2000.).

Color set for rescanning	Color set that segments RGB into 8 parts for each axis. This set contains 712 colors.	Color set that segments RGB into 10 parts for each axis. This set contains 1000 colors.	Color set that segments RGB into 12 parts for each axis. This set contains 1728 colors.
Strong Red of CIS (#b81c10) high saturation	 <p>6 colors</p>	 <p>13 colors</p>	 <p>19 colors</p>
Light Gray of CIS (#a1a194) low saturation	No color	 <p>2 colors</p>	 <p>3 colors</p>

The basic color set used for rescanning (henceforth CC) is a color set that segments the RGB axes. CC determines color-concept space. This color set is particularly suitable for high-performance rescanning. The color set for estimation (henceforth CCD) is a subset of CC. CCD is determined as the colors in the neighborhood of the specified colors. We evaluate the neighborhood distance  $D_n$  relative to the distance between pure red  $C_R$  and pure white  $C_W$  to separate from the color distance formula. The similarity  $S_l$  between the color  $C$  in CCD and the target color  $C_T$  is normalized by the neighborhood distance  $D_n$ . Then the neighborhood distance  $D_n$  and the similarity  $S_l$  are calculated as

$$D_n = \frac{\Delta E(C_R, C_W)}{N_0}$$

$$S_l = \begin{cases} \frac{D_n - \Delta E(C, C_T)}{D_n} & (\Delta E(C, C_T) \leq D_n) \\ 0 & (\Delta E(C, C_T) > D_n) \end{cases}$$

Where  $\Delta E$  is the color difference formula and  $N_0$  is the optimized divisor (1)

However, the number of colors in a neighborhood subspace differs depending on hue and saturation of the focused color. The number of colors in CCD for a low saturation color is very different from the number of colors in CCD for a high saturation color generally. As examples of the neighboring colors in the color set for rescanning shown in Table 5, when the neighborhood distance is one sixth of the distance between pure red  $C_R$  and pure white  $C_W$ , rescanning for Light Gray using 712 colors is invalid. Because there is no color in the centroid colors of 712 cubes in RGB color space within a neighborhood. In the same case for Strong Red, the calculation of

approximation is usually valid. Culture-dependent color sets are the same at this point. For example, there are one Japanese color, five Chinese colors, and eight French colors in the neighborhood of Light Gray in Table 5 under the same conditions. In this implementation, culture-dependent color definitions are due to DIC Color Guide. To ensure an appropriate resolution of color name estimation, the system dynamically optimizes the neighborhood distance and granularity of color quantization for the specified colors and purpose of estimation.

Table 6 shows an example of the optimization for Strong Purple-Blue (henceforth PB\_S) of CIS [19]. We used Japanese, Chinese and French color collections to define culture-dependent colors. The sets are Chinese (320 colors), French (322 colors), and Japanese (300 colors), as specified in DIC Color Guide [9]. The value displayed at the right side of the color name indicates the similarity  $S_l$  between the color  $C$  and the target color  $C_T$  (PB\_S) normalized by the neighborhood distance  $D_n$ . In this case the neighborhood distance is one fourth of the distance between pure red  $C_R$  and pure white  $C_W$ .

**Table 6.** The examples of the neighboring colors in the color set for rescanning around ‘Strong Purple-Blue’ (The neighborhood distance is one fourth of the distance between pure red and pure white. The color distance is calculated with CIE2000.).

Color Set for Color Histogram/ Color Definition Space	Japanese colors	Chinese colors	French colors	Color Set for Rescanning/ Color set that segments RGB into 8 parts for each axis. / Color Concept Space
2 colors	15 colors	18 colors	14 colors	21 colors
PB_S (1.00000)	コバルトブルー (0.66585)	深毛月色 (0.92585)	Bleu Roi (0.65075)	R01G03B05 (0.48087)
PB_DI (0.04088)	藍色 (0.51880)	浅海昌藍 (0.79939)	Lapis-Lazuli (0.61972)	R03G04B05 (0.45658)
	瑠璃色 (0.51114)	絨藍 (0.63924)	Bleu Bleu (0.59198)	R01G04B06 (0.45415)
	濃縹 (0.50474)	北京毛藍 (0.62365)	Bleu Mediterranee (0.58138)	R02G03B05 (0.45239)
	縹色 (0.38691)	沙青 (0.53833)	Pervenche (0.54084)	R02G04B06 (0.44075)
	紺青 (0.34540)	琉璃藍 (0.51496)	Royal Air Force (0.45059)	R02G03B04 (0.43017)
	紺碧 (0.23633)	海藍 (0.48910)	Bleu Acide (0.41765)	R01G03B06 (0.40657)
	鳩羽紫 (0.17806)	浅土藍 (0.40534)	Bleu De Chine (0.38922)	R02G03B06 (0.34254)
	露草色 (0.15351)	花青 (0.38383)	Bleu Gitane (0.36848)	R03G04B06 (0.33940)
	紺鼠 (0.13953)	鮮藍 (0.33941)	Gentiane (0.35293)	R02G04B05 (0.27634)
	濃藍 (0.12629)	羅藍灰 (0.24924)	Indigo (0.25142)	R01G03B04 (0.26074)
	中縹 (0.11094)	深竹月 (0.24763)	Bleu Royal (0.24012)	R01G04B07 (0.25137)
	熨斗目色 (0.10718)	鵝灰 (0.22381)	Bleu Turquoise (0.10414)	R02G04B07 (0.23080)
	藍鼠 (0.08925)	紺青 (0.22021)	Bleu Anglais (0.01192)	R01G03B07 (0.22306)
	鉛色 (0.07801)	群青 (0.19121)		R03G03B04 (0.19522)
		勞働布色 (0.18901)		R02G03B07 (0.15425)
		深毛藍 (0.12083)		R03G04B07 (0.09878)
		孔雀藍 (0.09109)		R01G04B05 (0.09376)
				R03G03B06 (0.08532)
				R03G03B05 (0.07074)
				R04G04B05 (0.04411)

The neighborhood distance is optimized to be the minimum distance to contain only two bins of color histogram definitions (first column of Table 6). The granularity

of color quantization is optimized to be the minimum segmentation number under the condition, that the number of colors in the CCD (fifth column of Table 6) is greater than the number of target culture-dependent colors in the neighborhood space (second - forth columns of Table 6). The example is optimized for three culture-dependent color sets for comparison.

### 4.3. Estimation of culture-dependent color names

Based on the recreated color histogram, the system estimates the most-similar culture-dependent color [12].

The CCD is a color set for estimation. A culture-dependent color  $c^t$  is transformed to a color distribution vector  $\mathbf{C}^t$  in CCD. The vector  $\mathbf{C}^t$  is calculated using weighting functions  $w(c, c^t)$  of  $c^t$  and a color  $c$  in CCD approximately. The vector  $\mathbf{C}^t$  in color-space CCD can be formulated as follows:

$$\mathbf{C}^t = (w(ccd_1, c^t), w(ccd_2, c^t), \dots, w(ccd_k, c^t))$$

where :  $w(c, c')$  is the weighting function (2)

and  $CCD = \{ccd_1, ccd_2, \dots, ccd_k\}$ .

To approximate traditional colors, the weighting function  $w$  can be defined as a function of color distance between traditional and CCD colors and interpreted as the similarity between two colors  $c, c'$ . We calculate the color distance using CIE2000. To calibrate the CCD color set, a constant number in the functions is assigned to allow similarity values of pure red and pure white to remain below the threshold value. The threshold is applied to each axis of color variations. The weighting functions can be defined by the following Gaussian functions:

$$\text{Gaussian function : } w(c, c') = \exp\left(-\sigma \cdot \Delta E(c, c')^2\right)$$




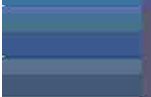
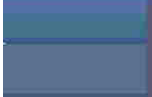
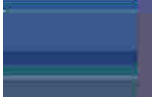
$$1 \geq w \geq 0 \quad w = 1 \text{ for the same two colors} \quad (3)$$

where  $\Delta E(c, c')$  : the color difference  
 $\sigma$  : constant for calibrating

The most-similar culture-dependent color name is estimated for the recreated color histogram. The similarity of the rescanned image histograms and transformed traditional colors is calculated in the neighborhood subspace of color concept space. Image feature vectors are projected on the traditional color vector by inner product, which needs some normalization. To normalize the weights of each traditional color, we normalize the vectors on CCD by every culture-dependent color using Euclidian distance.

Table 7 presents examples of the results of the estimation. The target images are from Ukiyoe collection and were published between 1821 and 1830. The three images in Table 7 are most similar to Strong Purple-Blue in CIS [19] used in each period. The third row shows the rescanned color distributions. The color set for rescanning is defined by segmenting RGB into 8 parts for each axis. This set contains 512 colors. The subset for estimation is constructed from 21 colors in the neighborhood of Strong Purple Blue. The neighborhood distance is one fourth of the distance between pure red and pure white. The color distance is calculated with the CIE2000 formula [20].

**Table 7.** Extracted culture-dependent color name for one focused color (Strong Purple-Blue) of the Ukiyoe images (1821-1835)

Period	1821-1825	1826-1830	1831-1835
Target images			
Rescanned color distribution			
Japanese	鉛色 (0.12283)	藍鼠 (0.58145)	コバルトブルー (0.18351)
Chinese	鵝灰 (0.10335)	花青 (0.58018)	浅海昌藍 (0.23631)
French	royal air force (0.09153)	bleu turquoise (0.07325)	royal air force (0.18253)
European	solitary blue (0.10420)	cloud blue (0.57941)	union jack blue (0.27067)

We used Japanese, Chinese, French and European color collections to define culture-dependent colors. The European Traditional Color set (1160 colors) was collected from Nicopon [23]. The other color sets are Chinese (320 colors), French (322 colors), and Japanese (300 colors), as specified in DIC Color Guide [9]. The value displayed on the right side of the color name indicates the similarity between the color distribution in the image and the color name. These color names represent subtle differences among three images for the sake of clarity.

#### 4.4. Aggregation and comparison of culture-dependent color names for image groups

Aggregation of the estimation results for each image indicates the characteristic colorization of images groups in detail. Visualization of the aggregation results enables the user to grasp the subtle differences regarding the specified colors.

The system has the following three aggregation function of culture-dependent color names.

- (1) A function to calculate the averaged area ratio of rescanning results
- (2) A function to calculate the estimation results of culture-dependent color names
- (3) A function to calculate the averaged similarity to culture-dependent color names

(1) is used for visualizing averaged area ratio with the color space for rescanning. These values indicate the area size normalized by image number and each image size. The function results are visualized with CCD colors. (2) is used for visualization of aggregation of most similar color name and similarity for each image. This represents both area ratio and similarity of culture-dependent color names. (3) is used for visualization of the averaged similarity to the estimated color, with normalization of the image number and the area size using the neighboring colors of focused colors. This indicates only the characters of color, not the area size.

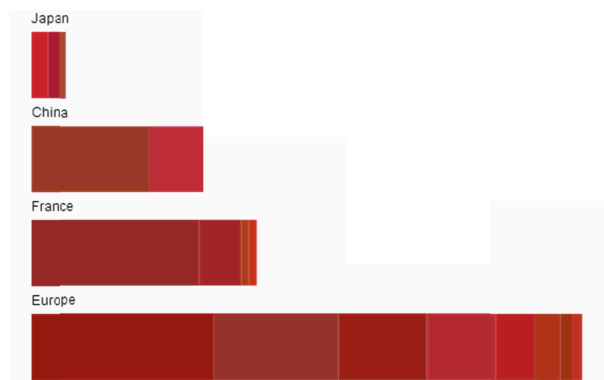
Visualization for color exploration is based on comparison. The system is able to show the comparison among groups and comparison among culture-dependent color definitions. In what following, the examples of comparison using these three functions are presented.

Table 8 shows the visualization results of the function (1) of rescanning images of Ukiyoe published during the period between Edo era and Meiji era (1864 to 1873). The images are grouped by year. Strong Red was made from imported aniline dye from the end of Edo period and was used widely after the Meiji Restoration (1868). The color set for rescanning is defined by segmenting RGB into 12 parts for each axis. This set contains 1726 colors. The subset for estimation is constructed from 21 colors in the neighborhood of Strong Red in CIS [19]. The neighborhood distance is one sixth of the distance between pure red and pure white. The color distance is calculated with the CIE2000 formula.

**Table 8.** Transition in Strong Red use in Ukiyoe published from 1864 to 1873 (the result of rescanning)

Year	Number of images	The averaged area ratio
1864	273	
1865	240	
1866	119	
1867	62	
1868	31	
1869	42	
1870	91	
1871	80	
1872	167	
1873	229	

Figure 5 shows the estimation results of the function (2) for Strong Red for the 8th row of Table 8 (items published in 1870) with culture-dependent color names. We used Japanese, Chinese, French and European color collections to define culture-dependent colors. The European Traditional Color set (1160 colors) was collected from Nicopon [23]. The other color sets are Chinese (320 colors), French (322 colors), and Japanese (300 colors), as specified in DIC Color Guide [9]. The figure visualizes the results of aggregation of similarity to the most similar color for each image. From this result, we can confirm that Strong Red in Ukiyoe published in 1870 is similar to European color name, because aniline dye was imported from Europe.



**Figure 5.** Estimation results for Strong Red in Ukiyoe published in 1870 with culture-dependent colors



Table 9 presents extracted results of the function (2). This culture-dependent color names for Strong Red in images of Ukiyoe published in 1870. The value displayed at the left of the color name indicates the number of images which contain the area of the culture-dependent color. The value on the right side of the color name indicate the amount of similarity between the color distribution in the image and the color name. These color names represent subtle difference.

**Table 9.** Extracted culture-dependent color names for Strong Red in images of Ukiyoe published in 1870

Japanese	Chinese	French	European
7 猩々緋 (0.21)	29 血紅 (1.54)	33 rouge sang (2.18)	14 henna (2.39)
7 深緋 (0.14)	15 象牙紅 (0.71)	13 piment (0.56)	14 brick red (1.62)
6 紅樺色 (0.10)		2 tomette (0.11)	9 brazil red (1.15)
		1 coq de roche (0.10)	12 bauk sweet (0.90)
			2 copper red (0.50)
			3 roastbeef (0.34)
			4 derby tan (0.15)
			1 tomato red (0.13)
			1 cardinal red (0.01)

Table 10 shows the result of the function (3). This shows averaged similarity between Strong Red and estimated culture-dependent color name in Ukiyoe published from 1864 to 1873. An averaged similarity is calculated from the aggregated similarity of estimated color with normalization of the image number and the area size used in the neighboring colors of Strong Red. These results indicate the transition of color, not the area size used the specified colors. Red color in Ukiyoe is similar to “紅樺色” in Japanese colors and “tometto” in French colors before 1870. After 1870, red color in Ukiyoe is similar to “猩々緋” in Japanese colors and “henna” in French colors. Thus, the system visualizes the subtle difference of specified colors in this way.

**Table 10.** Transition in averaged similarity between extracted culture-dependent color and Strong Red in Ukiyoe published from 1864 to 1873

year	number of images	The averaged similarity to cultural dependent colors		
		Japanese color names	Chinese color names	French color names
1864	273			
1865	240			
1866	119			
1867	62			
1868	31			
1869	42			
1870	91			
1871	80			
1872	167			
1873	229			

## 5. Conclusions

In this paper, we presented an explorative image analysis system and its application to comparative analysis of cultural arts and crafts. The main features of this system are the two explorative analysis methods with feature estimation and evaluation of culture-dependent colors: (a) image-group exploration and (b) color exploration. The system visualizes the distinct differences among image groups and provides notable images for users through the image-group exploration method. Furthermore, the system visualizes the subtle differences of colors in images by the color exploration method, with a zooming function for color distributions and cultural-color-name estimation.

We examined the feasibility and efficiency of our system by applying it to an Ukiyoe collection from the Tokyo Metropolitan Library. From the experiments, we confirmed that our system enables to extract the characteristic colorization of images and visualized the analyzed results on a time-series map and world map to provide an overview of the results.

This system will lead to a new image-exploration environment with deep analysis of color, which promotes better understanding of common features and subtle differences according to cultural background.

## References

- [1] Barakbah, A. R. and Kiyoki, Y., "An Emotion-Oriented Image Search System with Cluster based Similarity Measurement using Pillar-Kmeans Algorithm," *Information Modelling and Knowledge Bases*, IOS Press, XXII, pp.117-136, May 2011.
- [2] Bentley, Chris L., and Matthew O. Ward. "Animating multidimensional scaling to visualize n-dimensional data sets." *Information Visualization'96, Proceedings IEEE Symposium on*. IEEE, 1996.
- [3] Berlin, B. and P. Kay, "Basic Color Terms, Their Universality and Evolution," Berkeley and Los Angeles: University of California Press. First paperback edition 1991, with a bibliography by Luisa Maffi, 1969.
- [4] Chen, X. and Kiyoki, Y., "A Visual and Semantic Image Retrieval Method Based on Similarity Computing with Query-Context Recognition," *Information Modeling and Knowledge Bases*, IOS Press, Vol. XVIII, pp.245-252, May 2007.
- [5] Data-Driven Documents: <http://d3js.org/>
- [6] Deselaers, Thomas, Daniel Keysers, and Hermann Ney. "Features for image retrieval: an experimental comparison." *Information Retrieval* 11.2 (2008): 77-107.
- [7] Deselaers, Thomas. "Features for image retrieval." *Rheinisch-Westfälische Technische Hochschule, Technical Report*, Aachen (2003).
- [8] Dhiraj Joshi, James Z. Wang and Jia Li. "The Story Picturing Engine - A System for Automatic Text Illustration," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 2, no. 1, pp. 68-89, 2006.
- [9] DIC Graphics Corporation, DIC Color Guide: <http://www.dic-graphics.co.jp/en/index.html>.
- [10] Google Maps API: <https://developers.google.com/maps/>
- [11] Heider, E. Rosch, "Universals in color naming and memory," *Journal of Experimental Psychology*, 93, pp.10-20, 1972.
- [12] Itabashi, Y., Sasaki, S. and Kiyoki, Y., "Cross-cultural Image Computing with Multiple Color-Space Transformation," *Emitter Journal*, Vol.2 No.2, 2012.
- [13] Itabashi, Y., Sasaki, S. and Kiyoki, Y., "Distinctive-Color Analytical Visualization for Cross-Cultural Image Computing with 5D World Map," *Knowledge Creation and Intelligent Computing (KCIC 2013)*, March 20th – 21st 2013, South Bali, Indonesia.
- [14] Keogh, Eamonn, et al. "Intelligent icons: integrating lite-weight data mining and visualization into gui operating systems." *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, 2006.
- [15] Kiyoki, Y., Kitagawa, T. and Hayama, T. "A Metadatabase System for Semantic Image Search by a Mathematical Model of Meaning," *ACM SIGMOD Record*, 23, 4, pp.34-41, 1994.

- [16] Kiyoki, Y., Kitagawa, T., "A semantic associative search method for knowledge acquisition," *Information Modelling and Knowledge Bases VI*, H. Kangassalo et al eds. IOS Press 1995 pp. 121-130.
- [17] Kiyoki, Y., Sasaki, S., Nguyen N. T., Nguyen, T. N. D., "Cross-cultural Multimedia Computing with Impression-based Semantic Spaces," *Conceptual Modelling and Its Theoretical Foundations*, Lecture Notes in Computer Science, Springer, pp.316-328, March 2012.
- [18] Kiyoki, Yasushi, and Xing Chen. "Contextual and Differential Computing for the Multi-Dimensional World Map with Context-Specific Spatial-Temporal and Semantic Axes." *Information Modelling and Knowledge Bases XXV* 260 (2014): 82.
- [19] Kobayashi, S, "Color Image Scale, Kodansha International, 1991.
- [20] Luo, M. Ronnier, Guihua Cui, and B. Rigg. "The development of the CIE 2000 colour - difference formula: CIEDE2000." *Color Research & Application* 26.5 (2001): 340-350.
- [21] Nara Blog: <http://narablog.com/>.
- [22] Nguyen, T. N. D., Sasaki, S. and Kiyoki, Y., "5D World PicMap: Imagination-based Image Search System with Spatiotemporal Analyzers, " *Proceedings of IADIS e-Society 2011 Conference*, Avila, Spain, 8 pages, March 2011.
- [23] Nicopon, *World Traditional Colors*: <http://www.nicopon.com/iro/yo/>.
- [24] Niblack, Carlton W., et al. "QBIC project: querying images by content, using color, texture, and shape. " *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*. International Society for Optics and Photonics, 1993.
- [25] Open Source Computer Vision: <http://opencv.org/>
- [26] Sasaki, S., Itabashi, Y., Kiyoki Y. and Chen X., "An Image-Query Creation Method for Representing Impression by Color-based Combination of Multiple Images," *Information Modelling and Knowledge Bases*, Vol. XX, pp.105-112, 2009.
- [27] Shneiderman, Ben. "The eyes have it: A task by data type taxonomy for information visualizations." *Visual Languages*, 1996. *Proceedings., IEEE Symposium on*. IEEE, 1996.
- [28] Suhardijanto, Yasushi Kiyoki, Ali Ridho Barakbah, "A Term-based Cross-Cultural Computing System for Cultural Semantics Analysis with Phonological-Semantic Vector Spaces," *Information Modelling and Knowledge Bases XXIII*, pp.20-38, IOS Press, 2012.
- [29] The Musée de la Toile de Jouy: <http://www.museedelatoiledejouy.fr/>.
- [30] The Smithsonian American Art Museum: <http://americanart.si.edu/>.
- [31] The Tokyo National Museum: <http://www.tnm.jp/>.
- [32] Tokyo Metropolitan Library: <http://www.library.metro.tokyo.jp/>.
- [33] Tufte, Edward R. "Envisioning information." *Optometry & Vision Science* 68.4 (1991): 322-324.
- [34] Victoria and Albert Museum: <http://www.vam.ac.uk/>.
- [35] Wang, Yang and Acharya, "Color Space Quantization for Color-Content-Based Query Systems," *Multimedia Tools and Applications*, 13, 73-91, 2001.
- [36] Yang, Jing, et al. "Semantic image browser: Bridging information visualization with automated intelligent image analysis." *Visual Analytics Science And Technology*, 2006 *IEEE Symposium On*. IEEE, 2006.
- [37] Yang, Jing, et al. "Value and relation display for interactive exploration of high dimensional datasets." *Information Visualization*, 2004. *INFOVIS 2004. IEEE Symposium on*. IEEE, 2004.

# Adaptive Systems for Multicultural Deployment

Hannu JAAKKOLA <sup>a,1</sup> and Bernhard THALHEIM <sup>b,2</sup>

<sup>a</sup> *Tampere University of Technology, P.O.Box 300, FI-28101 Pori, Finland*

<sup>b</sup> *Christian-Albrechts-University Kiel, Computer Science Institute, 24098 Kiel, Germany*

**Abstract.** The paper develops an approach to information system adaptation supporting diversity and heterogeneity of users. The user perspective is handled from culture point of view. Cultures are layered and this structure can be used for starting with national cultures and deriving stereotypes. Culture is recognised as a multidimensional structure, in which national culture provides a basement for the behavioural variations of individuals in personal level. For information system development this layering covers three different perspectives: cultural stereotypes and user models, organisational models and technology models. We thus may develop a coherent information system perspective that supports adaptivity in multicultural information system applications.

**Keywords.** multi-cultural information systems, adaptation of systems to users, culture and user support

## 1. Introduction

### 1.1. *The Elements of Communication*

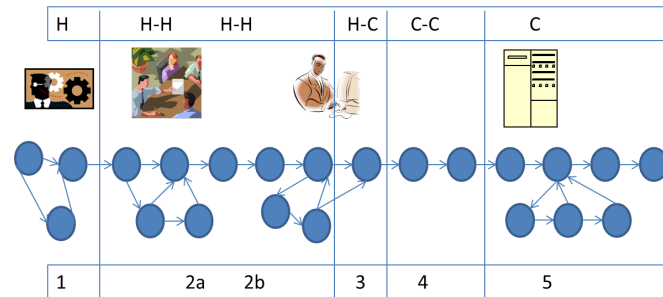
The use of information systems is interaction between people and computers. In addition to Human-Computer (H-C) interaction we can recognise other interaction types in the same flow of actions: Human reasoning (H), Human-Human Communication / Interaction (H-H), Computer-Computer Interaction (C-C) and Computing (C). All of these are relevant components in analysing the information system usage. Human to human communication represents the situation in which the problem is solved in the form of teamwork or the information system (IS) usage is delegated to the service person. In addition we can recognise human reasoning (H) and pure computing (C) as a part of the flow. Computer to Computer communication is needed when the service provided is based on collaboration (interaction) between different information systems (Figure 1).

Interaction is communication and communication is a workflow between the actors participating in it. The lower part of Figure 1 illustrates the communication as a workflow (dataflow). The data (connecting lines) is flown through transformations (circles) that modify the data (information chunks) in the flow. The responsibilities of each partici-

---

<sup>1</sup>hannu.jaakkola@tut.fi

<sup>2</sup>thalheim@is.informatik.uni-kiel.de



**Figure 1.** Communication in IS usage context

pating actor are marked as border lines. The border lines indicate the interface between the actors and specify the responsibilities of each actor. Successful interaction is guaranteed if the message transferred through the interface is understood in the same way by the sender and the receiver - i.e. the sender is adapted (processed the information) in the interface provided by the receiver. In H-H interaction the adaptation is usually based on common (natural) language or specifications that are understood in the same way by the sender and the receiver. In H-C interaction the interface is provided by the information system, as well as in C-C interaction; the difference is, however, in the existence or missing human intelligence.

### 1.2. The Research Problem and its Background

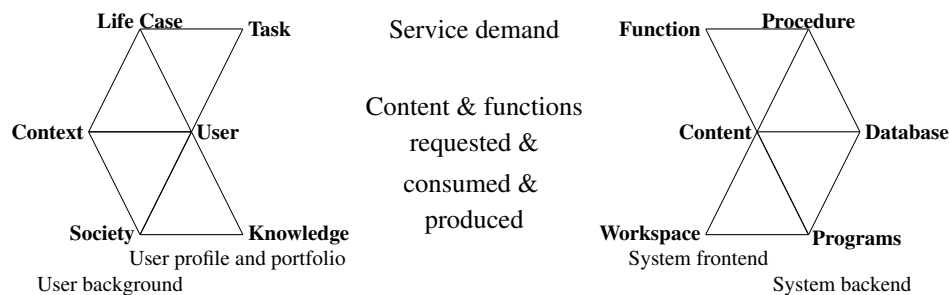
Our paper opens discussion on the development of information systems (IS) in a multicultural context. Two topics closely related to each other can be recognised: (1) what should be taken into account in developing information systems for heterogeneous group of users (the development process view), and (2) what properties in an information system support the heterogeneity in the use of them (information system view). Every day we meet problems and difficulties to use computer systems. There exists a conflict between the user's expectations and the real behaviour of the IS. The user interface represents the expected behaviour - the cognitive model - of the user in the H-C interaction. The difference between this cognitive model and the real behaviour of the user is the main reason for the problems. The reasons for these "usage" conflicts may be manifold: low quality of user interface (technical matter), lack of understanding of the users' real needs (inadequate requirements elicitation) and the heterogeneity of the users. In most cases the problems can be avoided by more careful software engineering work. The heterogeneity of the users is partially possible to solve by more careful engineering work, but in most cases, IS needs capability to adapt in the different needs of different users. Adaptability can be implemented in IS e.g. by different system configurations, by recognising the usage context and by learning capacity. These solutions are generally used in the situations, where the same user uses the IS with (logically) same user interface - e.g. in personalisation of mobile terminals (phone, pad).

Implementing adaptability in IS has several alternative approaches. The most common approach is based on the concept of "user modelling", which has its roots in human-computer research. In the paper [5] the user model is defined to represent a collection of

personal data associated with a specific user to be used as the basis for adaptive changes to the system's behaviour. This data may include personal information, users' preferences, data about their behaviour and their interactions with the system. Fisher lists four categories of user models. In static user models (1), once the user related data is gathered, it is not changed again. Shifts in users' preferences are not registered and no learning algorithms are used to alter the model. Dynamic user models (2) allow a more up to date representation of users. Changes are noticed and data related to it is updated to take the changed needs and goals of the users into account. Stereotype based user models (3) base on demographic statistics. Users are classified into common stereotypes and the system adapts to this stereotype. Highly adaptive (4) user models try to represent one particular user and therefore allow a very high adaptability of the system. In contrast to stereotype based user models they do not rely on demographic statistics but aim to find a specific solution for each user by recognising the different context related factors of the user.

### 1.3. The Exigency of the Research Topic

Classical information systems provide a set of user interfaces in a holistic one-world approach. Users may use the same interface set independent on their culture, their personalities and their specific needs. This situation is characterised in Figure 2. Information systems provide some content and functions for the usage of this content by users. Additionally, these systems may also support sessions by features such as workspace, e.g. sessions and specific storage facilities. The architecture of such systems can be based on a separation into frontend and backend.



**Figure 2.** The necessity for sophisticated information system support in dependence on the real demand of a given user

The user uses the system in dependence on the demands. Demands are driven by the life case and the general tasks a user has to solve. Life cases drive tasks that form the portfolio of a given user. Users have however their specific context, e.g. their general national and regional culture. Additionally users form on demand temporal collaborations and societies. Finally, users have their background, e.g. knowledge. We thus may characterise a user by their profile, i.e. the educational profile, the work profile, and the personality profile. This user profile is embedded into the society and the context of the user.

The classical service oriented architecture proposes that users are supported by a service detection facility, a service plug-in facilities and a service deployment facility. Since the resources for the development of such systems are limited, only a small number of services has been developed. Systems such as SAP are therefore rearranged and

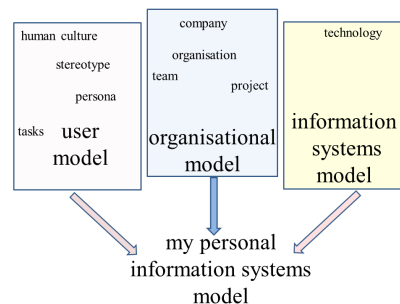
transfigured to the demands of their users by hundreds of small companies world-wide. This extensive approach results in hundreds if not thousands of interface suites for the same system.

#### 1.4. Research Directions and Our Approach

We discover thus that the user background, portfolio and profile is not properly taken into account in current technology. We may now derive a number of research tasks:

- Can information systems be partially adapted to specific user profiles and portfolio?
- Can we use user stereotypes for such adaptation?
- How can users be classified according to their culture and their specific behaviour?
- Is it possible to use this classification for system adaptation?

The answers to these questions should be positive if we want to have a sophisticated information system that pleases and satisfies the user. We observed that such systems must combine three models: the user model, the organisation model and the information systems model. Users should thus be provided by their personal information system as shown in Figure 3.



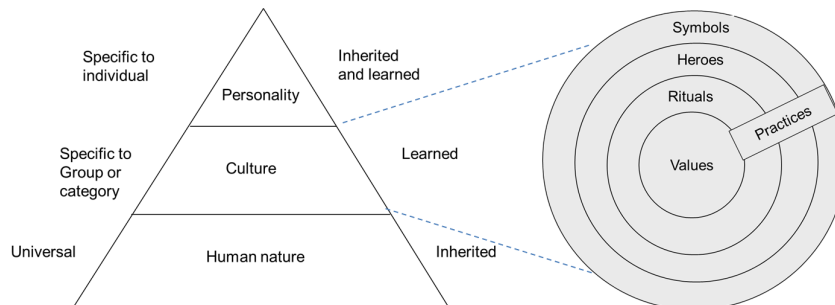
**Figure 3.** Combination of three models into a model for a personal information model

The paper shows that such combination is achievable. We first discuss in Section 2 two approaches for description of cultures and derive then stereotypes of such cultures. In Section 3 we derive how these stereotypes can be used for derivation of a user model. In Section 4 we develop technology solutions for such personalised information systems and conclude with a case study.

## 2. Understanding the Cultural Differences

### 2.1. The Layered Structure of Culture

At the beginning of his book Hofstede [8,9] defines the term culture as “*a collective phenomenon, which is shared with people who live or lived within the same social environment, which is where it was learned; culture consists of the unwritten rules of the social game; it is the collective programming of the mind that separates the member of one group or category of people from others.*” In this context he refers mainly in national culture and uses two models - pyramid model and onion model - to explain the layered structure of the culture and levels of uniqueness in mental programming (see Figure 4).



**Figure 4.** The layered structure of culture [9]

The lowest level of the pyramid (left part of Figure 4) is human nature. It is common for all human beings, universal and independent on group; it is inherited by birth and represents the “operating system” of the programmable human mind. The second level, culture, is specific to a specific group, “collective part of the program”. It is explained in more detail by onion model in the right hand side of the figure. The top level of the pyramid, personality, is specific to an individual. It includes both inherited and learned properties, which represents “personal mental programs”.

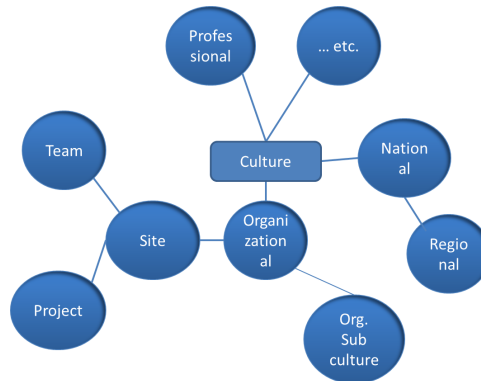
The onion model, right side of the figure, explains in more detail the elements of the culture. Values are the core of the culture and represent the preferred states over others. Rituals are collective activities related to a group of people; these rituals - like saying hello, ways of handshaking, etc. - indicate the togetherness of the members of the group and are also commonly accepted social norms of it. Heroes are characterising the highly prized persons as a model of behaviour in a culture. The outermost layer of the onion is symbols. These are words, gestures and objects that are common for those that share the culture. Practices are visible elements of the behavioural patterns of individuals. They are manifestations of the components having their source in different layers of the onion in the form of behavioural patterns, which are possible to interpret by knowing the roots of the elements. This forms also the basement for the stereotype based analysis of the cultural differences.

## 2.2. Multidimensional Characteristics of Culture

As discussed in the previous subsection, culture consists of the components that are partially inherited and partially learned. A human being is learning intensive and adaptable, which means that finally his / her culture consists of basic elements (as explained in Figure 4) and stock of lifeline experiences having their source in several different aspects (Figure 5).

*National culture* is the most dominant in the behaviour of individuals. Cultural aspects like language, educational tradition, religion, beliefs, attitudes, and social context are important sources of cultural dimensions too. It is also worth of recognising that national cultures have variations - called *regional cultures* (variations of the collective part of the software of the mind). *Organisational (work) culture* includes habits adopted by an organisation. It covers the similarities in behaviour, interaction, decision-making, organisational structure, and goals. People who have adopted the same organisational





**Figure 5.** Multidimensional aspects of culture

culture are able to communicate and transfer knowledge better than people from different work cultures. Analogical to national cultures, even organisational cultures may have variations: *site culture* covers the aspects typical to one operative site of an organisation; respectively projects and teams may develop their own variations. *Professional culture* has its roots in education and adopted practices typical to certain professions. *Project culture* and *team culture* are cross-sections of organisational and professional culture.

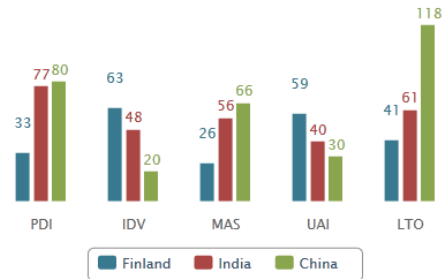
### 2.3. Cultural Stereotypes

*Cultural stereotypes* help to understand common patterns typical to national cultures. As discussed in earlier sub-sections national culture is the basement of the personality and explains a lot of the behavioural pattern of an individual. Most commonly used and referred frameworks that can be used to recognise cultural differences are published by Geert Hofstede [8,9,7] and Richard Lewis [14,15].

The model of Hofstede is based on the analysis of six cultural dimensions:

- *Power Distance* (PDI): the extent to which power differences are accepted;
- *Individualism / Collectivism* (IDV): the extent to which a society emphasises the individual or the group;
- *Masculinity / Femininity* (MAS): refers to the general values in the society - hard / soft values;
- *Uncertainty avoidance* (UAI): refers to the extent that individuals in a culture are comfortable (or uncomfortable) with unstructured situations;
- *Long-term / Short term orientation* (LTO): refers to the extent to which the delayed gratification of material, social, and emotional needs are accepted;
- *Indulgence / Restraint* (IVR): acceptance of enjoying life and having fun vs. controlling the life by strict social norms.

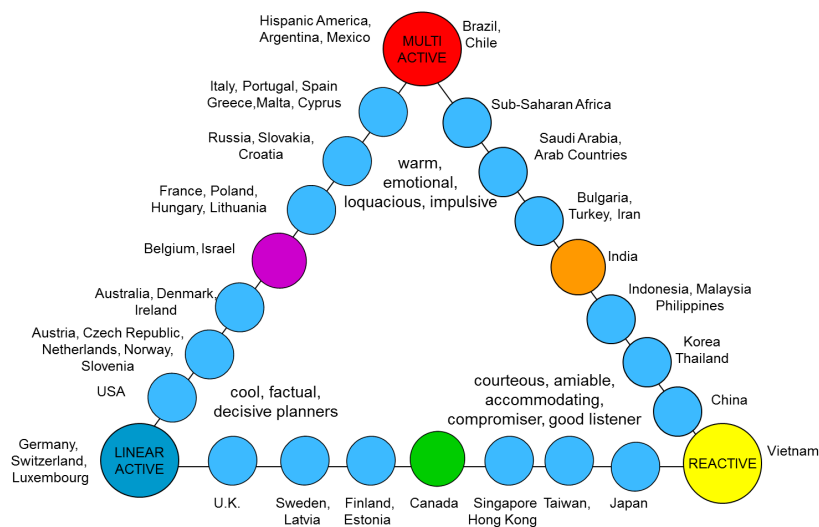
Hofstede's resource pages [7] provide a tool for visual comparison of selected national cultures. Figure 6 compares between national cultures of Finland, India and China; the two latter ones are most common target countries for offshoring and subcontracting in Finnish software industry and outsourced services.



**Figure 6.** Comparison of three national cultures applying Hofstede's analysis model

The comparison in Figure 6 indicates the most probable sources of cultural conflicts in organisational issues (PDI), leadership (PDI, UAI), decision making (PDI), organising the work (PDI, UAI), attitude to work (IVR, MAS), role of individuals in an organisation (IDV).

Lewis' model recognises three *basic stereotypes of cultures*. *Linear-active culture* is task-oriented and value is given to technical competence and facts. They are cool, factual and decisive planners. *Multi-active culture* is extrovert and human force is seen as an inspirational factor. They are warm, emotional, loquacious and impulsive. *Reactive culture* is people-oriented and dominated by knowledge, patience and silent control. They are courteous, amiable, accommodating, compromisers and good listeners. National cultures locate on the sides of a triangle having three basic stereotypes as angles (Figure 7).



**Figure 7.** National cultures in Lewis' model

The Lewis model gives detailed analysis of every national culture from four factors point of view: *general facts* (geography, history, politics and economy), *culture* (gen-

eral classification, values, cultural black holes, concept of time, concept of space, self-image), *communication* (communication pattern, body language, listening habits, audience expectation) and *interaction* (concept of status, gender issues, leadership, management, motivation factors, meetings, negotiating, contracts & commitments, manners and taboos, how to empathise). The country description available in internet resources, e.g.[15] is sufficient for analysis; additional information is available in several printed books.

The basic rule in applying Lewis' model in stereotype building is to look the cultural distance (distance in the triangle). If the basic stereotype category (edge in the triangle) is different (like in the case of Finland, China and India), to manage the problems rising up because of cultural differences is more challenging. The following short analysis clarifies the differences of these three cultures from different points of view.

*Culture classification:* Finland is reasonable linear-active culture, in which facts and data are important in opinion building. Tasks are executed in linear order one by one. Chinese and Indian people are reactive, Indians in addition reasonable multi-active people. Among them collective sources (internet, friends, family, and colleagues) of data have high importance in opinion building. Task execution is in these cultures based on multi-tasking (several tasks in one chunk) and prioritisation.

*Concept of space:* Finns feel themselves uncomfortable by any attempt to limit their personal space; face-to-face communication is at least 1.2 metres. Personal independency is in high value. Both China and India are crowd countries and people are used to live close together. These countries are typical representatives of collective cultures; this is confirmed also by Hofstede (Figure 6).

*Concept of time:* Finns are very punctual and value good time-keeping; time is divided and used for maximum efficiency. Indians have great latitude regarding punctuality. In China there is sense of the value of time. They frequently apologise taking up other people's time. They are punctual on arrival. On the other hand they expect a liberal amount of time to be allocated to repeated consideration of the details and to the careful nurturing of personal relationships surrounding the deal.

*Communication pattern:* Finns are different to the others. They are good listeners. They say only that which is absolutely necessary; they repeat and summarise. Body language is minimal or zero and facial expressions are limited. Indians communicate using lengthy and amiable small talk. They want hear the other side's view first; they have ability to modify and re-package it to achieve their own goal. Body language comes through eye contact and facial expressions. Chinese have courteous patient discourse. They attend meetings collectively and the real decision-makers are not actually in the meetings. Everyone protects everyone's face. Chinese do not usually look straight into someone's eyes when greeting.

*Listening:* Finns are world's best listeners and are trained not to interrupt. They give little or no feedback. They respect quality and technical information. Indians like flowery speech and an extensive vocabulary. They are willing to listen at length and to become a friend of the speaker. They respect know-how and trust. In China good listening is good manners. They accommodate the other side's wishes in their own proposals and acquire know-how from West. Trust creation is important.

#### 2.4. *The Concept of Culture Concluded*

The comparison between cultures provides understanding of the differences in behavioural patterns between people representing different cultures. The source of stereotype analysis is in different communication and collaboration context. It can be used to avoid and solve conflicts in typical conflict sensitive situations. In software engineering these are connected to communication, leadership, management (organising and division of the work), trust creation between collaborating parties, understanding the motivation factors and competence differences. In spite of having their main focus in work and organisations the models have also value in understanding the behavioural patterns of IS users. People's behaviour is guided, in addition to the national cultures, by several other cultural aspects, like organisational culture (habits adopted by the organisation), organisational subcultures and suborganisation cultures, work culture (similarities in behaviour, interaction, decision-making, organisation structure, and goals), professional culture (education and adopted practices typical of certain professions), project culture (cross-section of organisational and professional culture) and team cultures. Different aspects related to the globalisation in software engineering context are handled in the articles of the first author of this paper, e.g. [12,10,13,11].

### 3. **Towards Adaptive Information Systems**

#### 3.1. *Cultural Stereotypes and Information Systems*

M. D. Myers and F.B. Tan [17] discuss the role of national cultures in IS research. They suggest that in IS context the concept of "national culture" should recognise the emergent and dynamic nature of it; "It is something that is invented and re-invented and always in a state of flux". This points out an important aspect - people are pushed to change their traditions for several reasons and they are also all the time under the impact of foreign cultures. In the globalising world the culture is becoming all the time more and more global and it does not follow the borders of nations. Myers & Tan propose a research agenda for global information systems that takes seriously the idea that culture is complex and multidimensional and can be studied at many different levels. It can be studied at the international (e.g. West vs. East), national, regional, business, and organisational levels of analysis and these levels are often inter-connected and intertwined. The paper also criticises the use culture stereotype models because of simplicity. They equalise national culture and state culture, which is not right. In one administrative geographic area there may be several national cultures represented. There are also a lot of examples that people that are moved from their origins. They may live in the role of minorities in a foreign culture long time without losing their cultural identity. Because of that we argue that the stereotype based cultural models are beneficial to use in IS context. and that the dramatic changes happen more on the top level of the pyramid (Figure 4) - in personality level - without touching the kernel of culture and traditions (cultural stereotype). Instead, we fully agree with Myers & Tan that culture is multidimensional and also (in limited extent) dynamic concept. In our interpretation one important reason to the dynamics comes from the diversity of roles of individuals. Examples of roles are president of an international company, father of three small children, team leader of a local football team,

**Table 1.** Cultural factors in information systems

IS Property	Culture analysis
Complexity of UI	More accepted in high PDI than low PDI cultures. Multi-active cultures are used in multi-tasking and are more familiar in handling tasks in non-linear manner.
Long response times	More accepted in high PDI than in low PDI cultures.
UI colors	Accepted and expected in multi-active cultures. Color map and meaning of colors differs between cultures. In some cultures colors have also emotional connections.
Symbols and logos	In principle symbols provide a common language. Symbols in different cultures may have different meaning. There are also symbols that are not proper to be used in some cultures.
Support for uncertain situations	High UAI cultures respect features that guarantee the correctness of operations - e.g. repeating questions and extra confirmation operations. In linear-active those may be disturbing.
Decision making	In low PDI cultures the users are more ready to make fast decisions that have wide influence. In high PDI cultures the users would need confirmation from colleagues in higher positions, which leads to circulation of the activity step by step. This kind of circulation is felt inconvenient in low PDI cultures.
Privacy issues	Individualistic cultures are more aware of privacy issues than collective cultures.
Feedback	Fast feedback is appreciated in cultures that have low LTO level; high LTO index indicates readiness to wait for the feedback.
Clarity	Low UAI cultures are tended to accept confused situations in IS usage; high UAI cultures appreciate clarity.
Predictability	Predictability of information system's behavior is expected in high UAI cultures.

member of a Rotary Club. The individual changes his behavioural pattern according to the context. Every role based pattern is a combination of different cultural dimensions having different weights in different context. . In the following (Table 1) we give a short analysis of the possible cultural factors having affect in IS usage. The diversity of roles is discussed in sub-section 3.2 by introducing the concept of user models. User models provide means for variations in behavioral patterns in personal level, above the national culture level of the pyramid in Figure 4.

Table 1 gives some answers to the question “how cultural differences of the users should be taken into account in information systems”. The list is based on the subjective analysis of the authors derived from the stereotype analysis in Section 2; the aim is to give some examples for more detailed and careful analysis of the topic. The explanations to the behavioural differences are derived from cultural stereotypes discussed in Section 2.

### 3.2. User Models Refining the Stereotypes

Stereotypes of users can be combined into personae [16,20]. A *persona* is characterised by an expressive name characterising the stereotype, by culture, by nationality, by organisations or teams, by a bundle of projects the person might be engaged in, by a profession, by intents, by typical technical equipment, by behaviour pattern, by skills and profile, by disabilities, and by specific properties such as hobbies and habits. A persona is a typical individual created to describe the typical user, the context, the portfolio, and the profile. User models discussed below characterise profiles for education, work, and personality. This characterisation can be extended by an identity with name, pictures, etc., by personal characteristics such as age, gender, location, and socio-economic status, by a char-

acterisation of reaction to possible users error, by specific observed behaviour including skill sets, behavioural pattern, expertise and background, and by specific relationships, requirements, and expectations. A typical stereotype is the German Jack-of-all-trades that represents a specific kind of a business man in Germany.

User modelling is based on the specification of *user profiles* that address the characterisation of the users and incorporate the stereotype of the user as default values, and the specification of *user portfolios* that describe the users' tasks and their involvement and collaboration. Figure 8 can be extended to the more sophisticated model in [1].

To characterise the users of an information system we distinguish between *education*, *work* and *personality* profiles. The education profile contains properties users can obtain by education or training. Capabilities and application knowledge as a result of educational activities are also suitable for this profile. Properties will assigned to the work profile, if they can be associated with task solving knowledge and skills in the application area, i.e. task expertise and experience as well as system experience. Another part of a work profile is the interaction profile of a user, which is determined by his frequency, intensity and style of utilisation of the IS. The personality profile characterises the general properties and preferences of a user. General properties are the status in the enterprise, community, etc., and the psychological and sensory properties like hearing, motorial control, information processing and anxiety.

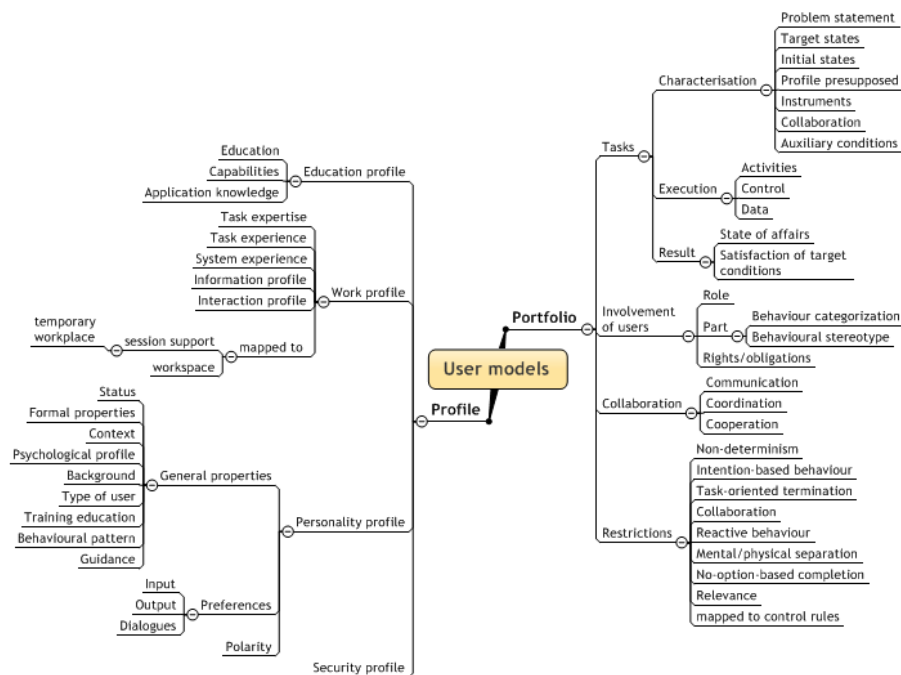


Figure 8. Profiles and portfolio of users

A *portfolio* is determined by responsibilities and is based on a number of targets. Therefore, the user portfolio within an application is based on a set of tasks assigned to or intended by a user and for which s/he has the authority and control, and a description

of involvement within the task solution [21]. A *task* as a piece of work is characterised by a problem statement, initial and target states, collaboration and presupposed profiles, auxiliary conditions and means for task completion. Tasks may consist of subtasks. Moreover, the task execution model defines what, when, how, by whom and with which data a task can be accomplished. The result of executing a task should present the final state as well as the satisfaction of target conditions.

### 3.3. Cultural Stereotypes, User Models and Information System Design

So far we derived the user model based on the cultural stereotypes. Classical information systems development is based on a three-layer architecture: the realisation layer, the conceptual layer and the user layer. The last layer is typically supported by a view set defined on top of the information system schema. We first develop a database schema and next derive a number of views on top of this schema. Using the schema and the views we may now apply transformation approaches for interpreting or translating the schema and the views to a realisation language, e.g. the data dictionary language provided by some object-relational DBMS.

This approach does not consider the culture or the user model. Based on our approach we derive now guidelines for the quality of the user interaction, guidelines for the development of the interfaces, and requests for adaptation of the system.

The view set must have also a clear and well-defined inner structure. This inner structure has to follow the logics of work, has to realise the specific requests according to the cultural factors in Table 1, and the user model. We may restrict for user interaction the complexity of the user interface, the response time, the feedback time and the colouring schema. Furthermore, we may require specific support for complex views, for decision making, for privacy, clarity and predictability. For instance for users within most German regional cultures, a complex user interface and a long response time should be avoided. At the same time, each region has its specific culture for colouring, texturing and layout. Additionally, features for supporting discussions, delay structures for later completion, support functions for privacy preservation have to be provided. Furthermore, we have to apply a number of interface criteria such as clarity of the layout and predictability for the control flow.

A guideline for the quality of user interaction includes:

*interface complexity* (tolerated, not tolerated): (level 1, level 2)  
*support* structure: view  
*support* discussion: view  
*support* functions: function  
*time behaviour* response time: maximal tolerated  
*time behaviour* feedback time: maximal tolerated  
*time behaviour* storage time: automatic  
*privacy* tactics: added system features  
*screen* guideline: description  
     *colouring* schema: description  
     *texturing* schema: description  
     *screen* quality criteria: description  
*workspace* personal: view  
     *additional workspace* personal: view  
     *shared workspace* (which user, which): (user,view)

This list is not exhaustive. It shows however how such guidelines can be derived for

certain user models. It allows however to satisfy typical requirements in relation to the personality of a user. It might be weakened in dependence on the education and work profile. It must however cover the portfolio of the user.

The user profile, the user portfolio and the corresponding stereotypes are also implicitly describing the properties of interfaces, e.g. ordering of items, effects that support the work, and the layout and playout of screens. Such guidelines are based on the principle of *proper organisation* depending on the user model, on the principle of *economy*, e.g. non-redundancy of actions, on the principle of *collaboration* depending on the skills and abilities of the user, and on system design *standards*. For instance, most regions in Germany use a clear, predictable and well-organised structure both for data (i.e. with proper layout) and the actions (i.e. proper playout).

In a similar form we derive *guidelines for the development of the interfaces*:

*organisation* structure: *pattern*  
*quality* maximal redundancy: *level*  
*supporting* abilities: *description*  
*support* structure: *view*  
*supporting* standards: *link*  
*layout* guide: *description of preferred*  
*playout* guide: *description of preferred*

This list is also not exhaustive and can be extended in dependence on habits and traditions and others.

Finally, interfaces and systems must be adapted to the user culture, to the specific regional culture, to the organisation and to the user themselves. The system specification may result in a large number of views and functions. We show however in the sequel that this set can be reduced by a clever realisation. Views and workplaces can be far too large, especially if users work on singleton screens. Therefore, we need an automatic decomposition feature for these features. We may base this decomposition on the principle of closeness of items, i.e. closed items are not separated and distant items can be separated. In a similar form we can organise the flow of work in dependence on the linearity of activities, the kind of multi-activity, and the inner structure of process deployment of users.

This set of properties can be combined to *requests for adaptation of the system* that have to be performed during transformation to a realisation system:

*separation* feature: *organiser*  
*based on* adhesion/cohesion: *sub-structures*  
*harmonisation* action set: *similarity level*  
*completeness* action net: *closure condition*  
*supporting* standards: *link*  
*quality* conciseness: *organisation preferred*  
*reactivity* support: *degree*  
*multi-activity* support: *pattern*  
*thought* direction: *linearity or network-oriented*

This kind of adaptation can be supported by current technology, e.g. by refinements of the system, by a larger set of views that support the user, by a larger set of functions, and by workplaces supporting the user.



## 4. Technology for Realisation of Adaptive Systems

### 4.1. View Towers for Information Systems

Information system design is often restricted to a three-level architecture. It is however folklore knowledge in real practical applications that views form their own architecture, i.e. views are defined on top of views that are defined on top ... of schema types. Therefore, *view towers* are already a common implicit folklore background. The research literature does not yet support such view towers. We do not know any publication on view sets after the systematic treatment of relational database development in [6]. Based on our projects we develop in the sequel a novel notion for view towers. This notion extends the notion of media types [19].

Database technology allows to define derived data on top of already obtained data. The derivation is based on the view definition. Views are defined incrementally on data dictionary types that have been defined before. Classically, (simple) views are defined as singleton types which data is collected from the database by some query:

```
create view NAME (PROJECTION VARIABLES) as
  select PROJECTION EXPRESSION
    from DATABASE SUB-SCHEMA
   where SELECTION CONDITION
   group by EXPRESSION FOR GROUPING
        having SELECTION AMONG GROUPS
   order by ORDER WITHIN THE VIEW;
```

The HERM [23] language and a HERM schema support a direct specification of a *view schema* by

- a schema  $\mathcal{V} = \{S_1, \dots, S_m\}$ , an auxiliary schema  $\mathcal{A}$  and
- a query  $q : \mathcal{D} \times \mathcal{A} \rightarrow \mathcal{V}$  defined on  $\mathcal{D}$  and  $\mathcal{V}$ .

Given a database  $\mathcal{D}^C$  and the auxiliary database  $\mathcal{A}^C$ . The view is defined by  $q(\mathcal{D}^C \times \mathcal{A}^C)$ .

Additionally, views should support services. Views provide their own data and functionality. The *group by*, *having*, and *order by* clauses provide a specific representation of the data generated by the view, i.e. by slice or rotate operations. These operations can be extended by dice, drill-down and roll-up functions [24]. We can use the same approach for an extension by functions that support browsing within the data, search, export, input and marking of data. Furthermore, we can use a workplace support by session functions. A *generalised view schema* [25] is given on the basis the frame:

```
generate MAPPING : VARS → OUTPUT STRUCTURE
  from DATABASE TYPES
  where SELECTION CONDITION
  represent using GENERAL PRESENTATION STYLE
        & ABSTRACTION (GRANULARITY, MEASURE, PRECISION)
        & ORDERS WITHIN THE PRESENTATION & POINTS OF VIEW
        & HIERARCHICAL REPRESENTATIONS & SEPARATION
  browsing definition CONDITION & NAVIGATION
  functions SEARCH FUNCTIONS & EXPORT FUNCTIONS & INPUT FUNCTIONS
        & SESSION FUNCTIONS & MARKING FUNCTIONS
```

Functions in a view are incrementally defined through expressions in the HERM algebra that incrementally use functions defined for already defined views and types. This specification is now to be extended by the deployment tactics:

```

authorisation USER LIST
  user OBLIGATION
    rights RIGHTS
      with grant OPTION
    roles USER WITH OBLIGATION
  read/update EXCLUSIVITY
enforcement PLACE AND TIME FRAME
  constraint SPECIFICATION
  visibility CONDITIONS FOR STABILITY AFTER UPDATE
    local/global ENFORCEMENT FRAME
  guarantees FOR VALIDITY
materialisation POLICY
  refreshment POLICY

```

The deployment tactic follows the capability relational DBMS are providing. The view is encapsulated and thus allows access only by authorised users with certain obligations within certain roles based on some transaction approach. View data might obey certain integrity constraints. Despite the view consistency we might also require global consistency to those views and data on which the view is build. View may also be materialised with some policy of direct refreshment or recharge if the underlying data are changed.

A *general view* combined generalised views with some deployment tactics. A *view tower* consists of a incrementally defined set of general views.

#### 4.2. *Adaptation and Layered Adaptive Systems*

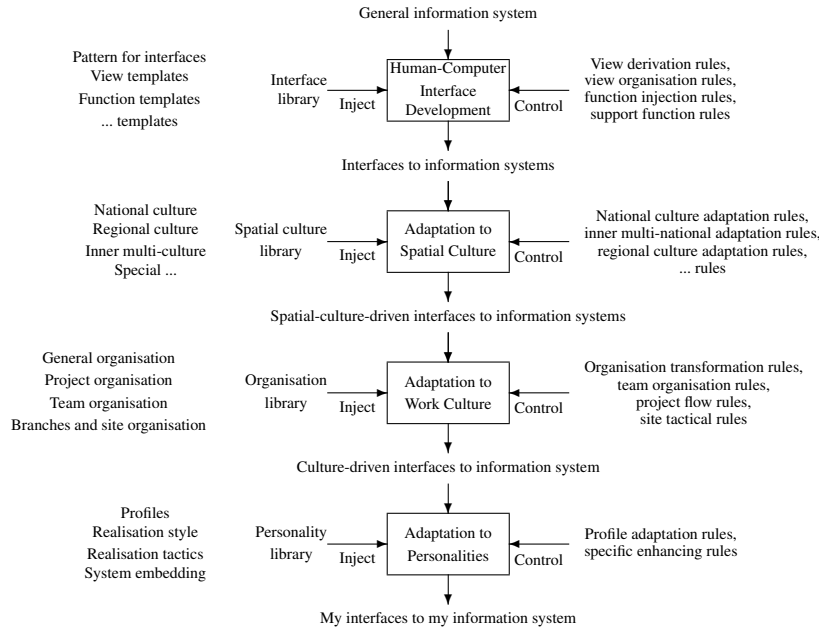
Classical information systems development mainly aims at development of the database schema, corresponding views and procedures and functions that might support an application. Adaptation to the user and their needs is sometimes provided by special programs.

Most systems today do not support adaptivity and user orientation. Information as processed by humans is perceived in a very subjective way. As for a knowledge system, the determining factor whether the user can derive advantage from the content delivered is the user's individual situation, i.e. the life case, user model and context. The same category of information can cause various needs in different life cases.

User typically request or need various content depending on their situation, on material available, on the actual information demand, on data already currently available and on technical equipment and channels on hand. This request is driven by the profile and portfolio of the user, by the national and regional culture, by the organisational and other cultures. Therefore, we need a facility for content and function adaptation depending on the culture and context of the user. Content adaptation and function adaptation may be thus considered as one of the 'grand' challenges of modern internet.

We observed already that culture may be layered into the general system landscape we use, into the national and regional culture and into different organisational cultures. Additionally, personality of a person must be taken into account. We use this layering for stepwise refinement of the view tower in Figure 9. Refinement of information systems can be based on ASM-like refinement [26].

Therefore, we develop first an adaptation to the spatial cultures, e.g. the national and regional culture. Next we may adapt the information system to the specific approaches that are used in organisations. We might have to revise the adaptation in the previous step



**Figure 9.** The stepwise adaptation of information systems to human use based on their demands, their spatial culture, their work culture, and their personality

if we have to consider multi-cultural organisations. Finally, we need to adapt the system to the specific personality of the user.

Database operations for view composition are join  $\bowtie$ , union  $\cup$ , selection  $\sigma$ , special selection or filtering against the data  $\otimes$ , projection  $\pi$ , rotation  $\rho$ , export cooperation with foreign databases  $\nearrow$ , import integration with data delivered from foreign databases  $\searrow$ , exclusive choice  $+$ , and exclusive split  $\Delta$ .

We base adaptation of a system on rules for transformation of views, functions and workspaces. These rules can be typically applied following the layering in Figure 9. They can however also applied in any order. The rule systems for adaptation must obey a Church-Rosser or the confluence property. These properties require that the application of one rule does not hinder the application of another rule. Since such a requirements is not satisfiable we require instead order-independence of a rule system which states that if a rule  $r_1$  is applied and another rule  $r_2$  is not applicable after this application then two another sequence  $r_{2,1}; \dots; r_{2,s}$  and  $r_{1,1}; \dots; r_{1,t}$  of rules must exist such that rule application sequences  $r_1; r_{2,1}; \dots; r_{2,s}$  and  $r_2; r_{1,1}; \dots; r_{1,t}$  have equivalent results.

A formal proof whether a rule system has this property is an open research issue. We can however develop context-free or local rules that have only an effect to local views, functions or workspaces. It is not yet our goal to develop such confluent rule systems. Instead we show how such a system can be build.

The adaptation of functions and the workplace will not be discussed in this paper. Since the workspace is based on the data and the functions then the adaptation of the workplace has to follow the adaptation of those. The adaptation of the functions may follow the BIER approach [22]. In this case, we should restrict refinement rules to local

rules that replace one separable unit by another one. If we restrict our rule system to *unit transforming rules* then we can use graph grammars [4] for adaptation rules.

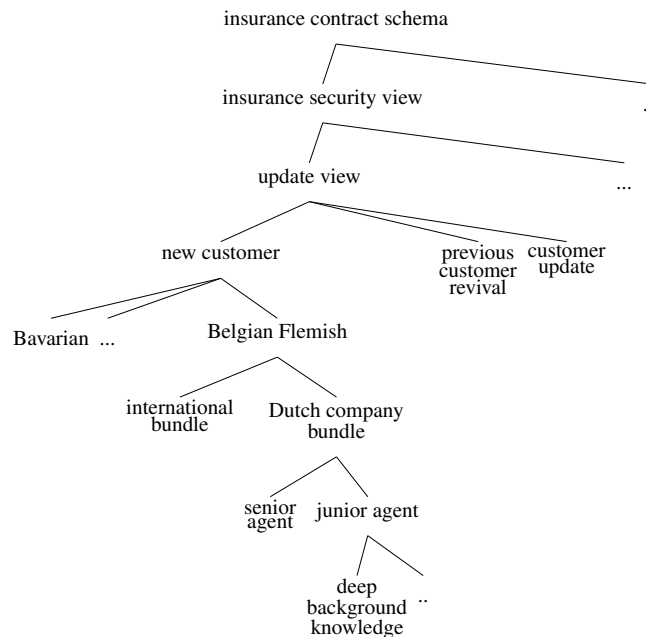
Our rule system uses the abstract state machine (ASM) approach [3]. It uses the pattern

if CONDITION then ACTIONS

in which conditions can be event- or data-based as well as control-driven and actions are allowed on the entire system specification as long as this rule application does not conflict with other concurrent application of other rules. In this case an adaptation can be understood as a refinement [2,18] of the system, i.e. the new system behaves on the specific restricted scope in the same way as the old system within the same scope.

#### 4.3. A Case Study

Systems for field staff are typical examples of cultural-dependent, multi-faceted, multi-structured and multi-functional systems. They must provide many interfaces in order to satisfy all the demands. Database technology supports however such systems based on view towers, e.g. the incremental structure in Figure 10<sup>3</sup>. The final view of junior field



**Figure 10.** A part of the view tower used in insurance applications for field staff agents

staff agent depends on the educational, work and personality profile of the given agent. The agent has to follow the organisational, project and team culture depending on the

<sup>3</sup>The views until level 3 have been developed in an industrial project of the second author. During this project we, however, realised that the development of views until level 7 would be very beneficial. Although the system developed is still in deployment there are many requests for changes that might provide a better agent support within this application.

bundle of companies collaborating. Customer support depends on the region and the culture within the given region. The upper level views follow the typical structure of view-backed information systems. Work views enhance update etc. views by additional data and features. Update views allow a direct update through this view in the given database. The data are gated through the security views directly to the database which is structure according the database schema.

Let us restrict the case study to the general view that directly supports the negotiation of field staff agents with the customer. It forms a complex HERM schema. Let us denote the customer view by  $v_1$ , the activity supporting views by  $v_2$  (revival supporting view:  $v_{2,1}$ ; update tracking view:  $v_{2,2}$ , new customer supporting view:  $v_{2,3}$ ), collaborating companies views by  $v_3$ , agent profile views by  $v_5$  (education view:  $v_{5,1}$ , work profile view by  $v_{5,2}$ , responsibility view for field staff  $v_{5,3}$ ),

Additional views that provided by the spatial culture, organisation and personality libraries are national and regional customer culture views  $v_{16}$  and  $v_{21}$ , views for preparation of records or contracts  $v_{10}$   $v_{11}$ , views for contracts with the customer  $v_6^{Document}$ , an auxiliary view for enabling the contracting with supporting companies  $v_9$ , views for preparation of records or contracts  $v_{10}$   $v_{11}$ , consultation view with other departments in the company by  $v_{17}$   $v_{18}$ ; proposal views by  $v_{25}$ , and forms view for proposals  $v_{5,2}^{Prop}$ .

We use adornment indexes for views that are directly defined on its component. The left lower adornments are used for the regional culture, e.g.  $BF$  and  $GB$  for the Belgium-Flemish and Germany-Bavarian. The collaboration view with supporting companies typically requires a record of the negotiation policy, e.g. by a view that supports start of negotiations, the negotiation itself and a concluding summary  $Start_{v_{3,1}}$   $\bowtie$   $Nego_{v_{3,2}}$   $\bowtie$   $Concl_{v_{3,3}}$ . Such negotiation is also followed by an injection of some formal contract extension view  $Form_{v_3}$ . The right upper adornment is used for shortcuts such as an identifier based compilation of all prerequisites, e.g.  $v_9^{ID}$ .

The starting view is the customer view:

- $v_1 \bowtie ((v_{2,1} \cup v_{2,2}) \Delta v_{2,3}) \bowtie v_3 \bowtie (v_{5,1} \cup (v_{5,2} \bowtie v_{5,3}))$

This view must however be adapted to the national and regional culture, to the specifics of treatment within the given insurance company and to the profile of the agent. Let us consider the case of adaption for the view component  $v_3$ , i.e. for the view that provides data about supporting companies. The view tower for the application can now be composed based on the layering in Figure 9 as follows:

- We apply rules that add to customer view data that characterise prerequisites for involving a supporting company and adapt to the national culture:

$$\frac{v_{16} \bowtie [v_{21} \bowtie] v_1 \bowtie ((v_{2,1} \cup v_{2,2}) \Delta v_{2,3}) \bowtie v_9 \bowtie v_3 \bowtie (v_{10} \cup v_{11}) \bowtie}{(v_{5,1} \cup (v_{5,2} \bowtie v_{5,3}))}$$

- Now we can apply rules that allow to consider regional culture, insurance policy and support proposal:

$$\frac{v_{16} \bowtie [v_{21} \bowtie] v_1 \bowtie ((v_{2,1} \cup (\frac{BF}{v_{2,2}} + \frac{GB}{v_{2,2}})) \Delta v_{2,3}) \bowtie [(\nearrow v_{17} \bowtie v_{18} \searrow \bowtie)] v_9 \bowtie \frac{Start_{v_{3,1}} \bowtie Nego_{v_{3,2}} \bowtie Concl_{v_{3,3}} \bowtie Form_{v_3} \bowtie (v_{10} \cup v_{11})}{(v_{5,1} \cup (v_{5,2} \bowtie v_{5,3}))}$$

- Since empty parts of the views are not of interest we apply a rule for filtering the generalised view against the current one:

$$v_{16} \bowtie [v_{21} \bowtie] v_1 \bowtie (\otimes_{BF} v_{2,2} \otimes \Delta v_{2,3}) \bowtie [(\nearrow v_{17} \bowtie v_{18} \searrow \bowtie)] v_9 \bowtie \frac{Start_{v_{3,1}} \bowtie Nego_{v_{3,2}} \bowtie Concl_{v_{3,3}} \bowtie Form_{v_3} \bowtie (v_{10} \cup v_{11}) \bowtie (\otimes v_{5,2} \bowtie v_{5,3})}{(v_{5,1} \cup (v_{5,2} \bowtie v_{5,3}))}$$

- Now a general organisation rule is applied. An insurance contract number covers the insurance history:

$$\frac{v_1^{Hist} \bowtie [(\nearrow v_{17} \bowtie v_{18} \searrow \bowtie)] v_9^{ID} \bowtie \text{Start}_{v_{3,1}} \bowtie \text{Nego}_{v_{3,2}} \bowtie \text{Concl}_{v_{3,3}}}{\bowtie \text{Form}_{v_3} \bowtie (v_{10} \cup v_{11}) \bowtie (v_{5,2} \bowtie v_{5,3})}.$$

- Finally we cope with negotiation history, additional proposals and forms to fill:

$$\frac{v_1^{Hist} \bowtie v_{25} \bowtie [(\nearrow v_{17} \bowtie v_{18} \searrow \bowtie)] v_9^{ID} \bowtie \text{Start}_{v_{3,1}} \bowtie \text{Nego}_{v_{3,2}} \bowtie \text{Concl}_{v_{3,3}}}{\bowtie \text{Form}_{v_3} \bowtie (v_{10} \cup v_{11}) \bowtie (v_{5,2} \bowtie v_{5,3}) \bowtie \underline{\underline{v_6^{Document}}}}.$$

The final view can easily be decomposed into a stream of associated views based on the technology developed for media types [19].

The functions and the workplace of the agent is adapted in a similar way.

## 5. Conclusion

Software systems are often not adaptive to the user since a one-interface-for-everybody approach has been used. Such systems must be learned by users before they can properly use the system. It is however possible to develop adaptive information systems. This paper discusses one potential solution. We model the culture from the national, regional and organisational side. These models allow to derive stereotypes of users. These stereotypes can be combined with the profiles of users and with portfolio of users. Information technology allows to define views on top of the database system. These views can be layered in such a way that each layer adapts to one of the next facets. Interfaces are then assigned to each kind of user. In this paper we restrict the construction to the data side.

The development of the same kind of adaptation features for functions and workflows can follow the same path as long as these workflows can be conservatively extended and as long as these workflows have a Fitch structure. Then we may restrict the adaptation locally to those parts that must be adapted. The development of the workspace adaptation features is a combination of the data and the workflow adaptation.

A good number of open research problems has still to be solved: confluence of rule application throughout the adaptation process; stereotypes of users; collaboration in a multicultural setting; quality characteristics of multicultural information systems; assessment of adaptive information systems; capability of culture-sensitive systems; development of benchmarks and testbenches; coherence of adaptation rules; development of multicultural pattern; development of guidelines for flexible adaptation.

## References

- [1] M. Altus. *Decision support for conceptual database design based on evidence theory - An intelligent dialogue interface for conceptual database design*. PhD thesis, Brandenburg University of Technology at Cottbus, Faculty of Mathematics, Natural Sciences and Computer Science, 2000.
- [2] E. Börger. The ASM refinement method. *Formal Aspects of Computing*, 15:237–257, 2003.
- [3] E. Börger and R. Stärk. *Abstract state machines - A method for high-level system design and analysis*. Springer, Berlin, 2003.
- [4] H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, editors. *Handbook of graph grammars and computing by graph transformations. Vol. 2: Applications, languages and tools*. World Scientific, Singapore, 1999.
- [5] G. Fischer. User modeling in human-computer interaction. *User Model. User-Adapt. Interact.*, 11(1-2):65–86, 2001.

- [6] C. C. Fleming and B. von Halle. *Handbook of relational database design*. Addison-Wesley, Reading, MA, 1989.
- [7] G. Hofstede. Cultural dimensions - WWW. <http://www.geert-hofstede.com>, Retrieved November 20th, 2013.
- [8] G. Hofstede and G.J. Hofstede. *Cultures and Organizations: Software of the Mind: Intercultural Cooperation and Its Importance for Survival*. McGraw-Hill, New York, 2004.
- [9] G. Hofstede, G.J. Hofstede, and M. Minkow. *Cultures and Organizations: Software of the Mind: Intercultural Cooperation and Its Importance for Survival*. McGraw-Hill, New York, 2010.
- [10] H. Jaakkola. Towards a globalized software industry. *Acta Polytechnica Hungarica*, 6(5):69–84, 2009.
- [11] H. Jaakkola. Culture sensitive aspects in software engineering. In *Conceptual Modelling and Its Theoretical Foundations*, volume 7260 of *Lecture Notes in Computer Science*, pages 291–315. Springer, 2012.
- [12] H. Jaakkola, A. Heimbürger, and J. Henno. The roles of knowledge and context in context-aware software engineering - in terms of education and communication. In *MIPRO 2009*, pages 224–230, 2009.
- [13] H. Jaakkola, J. Henno, and P. Linna. From local to global - path towards multicultural software engineering. *International Journal of Knowledge and Learning*, 2011.
- [14] R.D. Lewis. *When Cultures Collide. Managing Successfully Across Cultures*. Nicholas Brealey, London, 3rd edition, 2011.
- [15] R.D. Lewis. Richard Lewis resource pages - Cross-culture. <http://www.crossculture.com/services/cross-culture/> & <http://www.cultureactive.com>, 2013. Retrieved November 20th, 2013.
- [16] S. Mulder and Z. Yaar. *The User Is Always Right: A Practical Guide to Creating and Using Personas for the Web*. New Riders, Berkeley, 2006.
- [17] M. D. Myers and F.B. Tan. *Advanced topics in global information management*, chapter Beyond models of national culture in information systems research, pages 14–29. IGI Publishing, 2003.
- [18] G. Schellhorn. ASM refinement preserving invariants. *Journal of Universal Computer Science*, 14(12):1929–1948, 2008.
- [19] K.-D. Schewe and B. Thalheim. *Web Information Systems*, chapter Structural media types in the development of data-intensive web information systems, pages 34–70. IDEA Group, 2004.
- [20] K.-D. Schewe and B. Thalheim. User models: A contribution to pragmatics of web information systems design. In *WISE*, volume 4255 of *Lecture Notes in Computer Science*, pages 512–523, 2006.
- [21] K.-D. Schewe and B. Thalheim. Development of collaboration frameworks for web information systems. In *20th Int. Joint Conf. on Artificial Intelligence, Section EMC07 (Evolutionary models of collaboration)*, pages 27–32, Hyderabad, 2007.
- [22] M. Schrefl and M. Stumptner. Behavior consistent refinement of object life cycles. In *ER*, volume 1331 of *Lecture Notes in Computer Science*, pages 155–168. Springer, 1997.
- [23] B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.
- [24] B. Thalheim. The enhanced entity-relationship model. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 12, pages 165–208. Springer, Berlin, 2011.
- [25] B. Thalheim. Web information systems: Analysis, design, development, and implementation of business sites, collaboration sites, edutainment (e-learning) sites, and infotainment (information) sites. <http://www.is.informatik.uni-kiel.de/thalheim/WIS.pdf>, 2013. Assessed Oct. 26, 2013.
- [26] Q. Wang and B. Thalheim. Data migration: A theoretical perspective. *DKE*, 87:260–278, 2013.

**Acknowledgment.** We would like to thank the Academy of Finland and the German Academic Exchange Service (DAAD) for the support of this research.

# Text-retrieval in SQL and No-SQL environments

Jevgenij JAKUNSCHIN <sup>a,1</sup>, Antje DÜSTERHÖFT <sup>a</sup> AND  
Christoph EIGENSTETTER <sup>a</sup>

<sup>a</sup> *Faculty of Engineering, University of Wismar, Germany*

**Abstract.** Big data (e.g. social media data) requires new approaches of text processing. This article covers the evaluation of No-SQL and SQL databases (SAP HANA, Oracle, HBASE), while focusing on different indexing options, full text search types, search accuracy, performance and possible semantic and information modeling options. This includes the generation of an adequate test data collection, functionality tests and the evaluation of syntax and semantic modeling capabilities.

**Keywords.** databases, full text processing, SAP HANA, NoSQL, In-Memory databases, text mining

## Introduction

In the past decade, the amount of data, companies and projects have to manage has grown exponentially. New requirements lead to new approaches and several new database subtypes were developed, specifically No-SQL and In-Memory databases. In contrast to the SQL language, there is no common language for No-SQL databases yet. In this progressively developing environment non-core functions are often not present or lack functionality. A common example is full-text functionality, which requires substantial processing power.

The goal of this work is the comparison and evaluation of SQL and No-SQL environments. The primary focus is set on full-text functionality and performance. The SAP HANA environment was chosen because of the big number of full text processing methods, the in-memory database nature and a diversity of multi-threading optimizations. This article evaluates multiple criteria including performance, full-text functionality, system stability and format compatibility. The test data is extracted from two different sources: a merged table of over 15000 books and a twitter extraction application.

Performance and optimization tests are performed on further SQL and No-SQL systems afterwards. At the point of writing, this project is still work-in-progress. The final results can be found in the paper, by the end of May 2014[1].

Finally different setups are compared to display the viability of the tested systems for full-text management operations.

---

<sup>1</sup>Corresponding Author: Jevgenij Jakunschin, University of Wismar; E-mail: j.jakunschin@stud.hs-wismar.de



## 1. Related Work

An overview about techniques for No-SQL databases can be found in ([12],[13],[14]). The presentation[6] compares the performance and stability of 4 No-SQL databases, while also provides insight on their backup strategies and statistics as well as database behavior in case of a server failure.

An example of an effective implementation of a full text retrieval systems with SAP HANA is the Springerlink system[16].

In [15] and [7] a wide variety of information retrieval techniques, especially focusing on text and databases are provided.

## 2. Project details

The following strategy was used to determine the full text processing efficiency of different databases and achieve the necessary optimizations.

First, by observing several examples, we determined the relevant text-types to fill the database with. The next step was to find effective ways to generate or extract similar samples of test data and develop applications to perform this task. This was done in order to preserve the practical relevance of the work. Several databases were analyzed and a selection of systems was chosen, each representing certain approaches and aspects.

However, each database employs different data standards and requirements and features different optimizations. In order to take advantage of this aspect, additional applications were designed to manipulate the data and adapt it's form to the required standards. These manipulations range from changing the data encoding, to generating additional errors in the text up to test specific full-text functions (eg. fuzzy search efficiency).

Once the test data and the applications were designed, the right hardware was picked. Full-text operations often require substantial processing power, hence the hardware choice is a non-trivial task. Using the same system installation for different databases allows a simpler comparison of the selected database performance. The next step was to decide the right performance tests. This includes indexing options, search types and optimizations. Once the tests are performed, the results are evaluated and compared to similar results from the other systems.

### 2.1. Test systems

The following systems were picked as primary testing environment:

1. Oracle - The Oracle database is a popular relational database system with a high variety of full-text features, highly optimized data structures and the capability of processing "big data" information
2. SAP HANA - The in-memory database SAP HANA features several full-text processing methods, both row and column store, storage capabilities, multi-threading optimizations as well as data mining and analytic modules.
3. HBASE - The HBASE environment is a popular No-SQL database system, equipped with a wide array of full-text processing options.

## 2.2. Test data

One requirement to successful evaluation of text management systems is an optimized collection of texts. Non-optimized data-sources can cause compatibility issues with certain databases, special characters can interfere with CSV file formatting and more complex encoding systems will decrease the system performance or can generate wrong characters if unsupported. In addition, we need substantial amounts of text to reproduce big data scenarios and fully load big server clusters. To find the right text collection was quite challenging.

Different full-text environments and methods are optimized for different types of data. Some are optimized for real-time tasks, while other methods are used in data mining and analytic applications. Data sources were evaluated by the following criteria: quantity (in order to fully load the system), language (single language texts are preferred), data purity (avoiding special characters, HTML encoding and other undesired data), text type, text format (HTML, TXT, PDF...) and the possibility to expand the data collection.

These requirements contain some contradicting issues. Providing large quantities of consistent data, while maintaining the possibility to expand the collection with real time data from a single source is complex. Data sources with high expandability (eg. Twitter feeds) often provide a consistent text format, at the cost of language and data purity.

Considering the fact that different tests often require different data qualities, we decided to collect data from 2 different, separate sources, each representing it's own qualities and advantages:

1. The Project Gutenberg [10] offers thousands of e-books for download. The books differ in size, language, release year and format. The website also offers options to programmatically extract the books, while filtering by language, file type, year and other options. This data source provides high quantities of texts with consistent formats, text types, purity and language. The downside is the low expandability and the inconsistent text size and structure.
2. A Twitter Crawler [11] (twitter feed extraction application) downloads and saves tweets about a specified topic in real-time. The application is able to quickly collect small amounts of additional data, but the data is not as pure compared to the Gutenberg data source. However, the twitter crawler provides an effective method to gather a large collection of real-time data with a consistent data size.

Certain tests require the data to have a certain format or size. Other evaluation types require a certain average text-per-cell amount to be effective. In order to provide the optimal conditions for performance evaluation, we also designed an application that allows effective restructuring of text data.

An example would be a function that can insert intentional, controlled errors into the test data to evaluate the effectiveness of different fuzzy search algorithms and their performance cost. Another additional feature converts high amounts of text files into CSV files. It also allows to optimize texts for different databases, tests and preserve consistency by changing encoding and removing special characters, that might cause conflicts with the database and with the CSV delimiting characters.

Further functions include changing the amount of text data saved per row for indexing tests. This allows to save 1 book / cell or 1 paragraph / cell or a fixed amount of text.

The application can also add additional columns that contain special information, useful for certain tests like: line count, text length and book data.

### 2.3. *Hardware setup*

To test the full text functionality of different database environments on an industrial level, we include operations with large scales of data, that require hardware capable of processing "big data" problems. In addition, different systems are optimized for different hardware setups. Certain systems prefer a network of servers, while others perform best on a single server. In addition "in-memory" systems require a high amount of available RAM.

The system chosen for the project is the original SAP HANA server setup. These servers feature 40 cores, providing sufficient processing power to run performance tests with high quantities of text data. That means, we also have the possibility to test multi-core optimization. A particular operation can be executed with 1, 2, 4, 8, 16, 32 and all available cores and the performance boost can be measured.

The hardware provides 1 terabyte of RAM, which allows in-memory database tests.

Due to differences in hardware requirements, specific software-designed hardware and other problems, it was not possible to run all 3 databases on the same hardware. In order to overcome this hinderance this project is using a number of different approaches and evaluation factors to compare the different systems.

### 2.4. *System tests*

In order to create a comprehensive analysis of system capabilities we take the following approach.

The first step is to find and test all common text retrieval techniques between the three systems. This is done to create a common comparison ground. Methods exclusive to one database will also be evaluated, but usually most databases support a set of common methods like: "regular select/search", "full text exact search", "full text fuzzy search", "full text boolean search", "wild-card search".

Subsequently, the test data is imported into all tested database systems. Multiple tables with the test data are created to apply different indexing strategies.

Each full text retrieval technique is used with each table in each of the three systems. Once the fastest pair is selected on each system, we test the selected pair with more parameters - multi threading efficiency (eg. data import), argument influence (eg. fuzzy search), additional indexing options (eg. Boolean search) etc.

Finally the most effective methods and configurations are double-checked (with a different data set) and compared to the other systems and approaches.

The results are evaluated afterward. It should be considered, that the used hardware was optimized for the SAP HANA system, so results might favor in it's direction. Thus, we try to provide optimization strategies and point out advantages of different setups and discover tendencies (exponential growth, linear growth, multi-threading efficiency, indexing strategies) rather than simply point out which system performed the task quickest.

### 3. Conclusion and future work

This article described strategies to compare different database systems with a specific focus on big data text-mining and the differences of SQL- and No-SQL-databases. The evaluation and suggestion of use cases based upon the data acquired during the project can be found in the full document

### References

- [1] Jakunshin, Jevgenij: Text Retrieval in No-SQL and In-Memory Databases, 2014
- [2] Berg, Markus M., Isard, Amy and Moore, Johanna D.: An OpenCCG-Based Approach to Question Generation from Concepts. In: Natural Language Processing and Information Systems, pages 38-52, Springer Berlin Heidelberg, Lecture Notes in Computer Science 7934, 2013
- [3] Markus Berg, Antje Düsterhöft and Bernhard Thalheim: Towards Interrogative Types in Task-oriented Dialogue Systems. In: 17th International Conference on Applications of Natural Language Processing to Information Systems, Springer, Groningen (The Netherlands), Lecture Notes in Computer Science 7337, 2012
- [4] Markus Berg, Antje Düsterhöft and Bernhard Thalheim: Query and Answer Forms for Sophisticated Database Interfaces. In: 22nd European Japanese Conference on Information Modelling and Knowledge Bases, Prague (Czech Republic), 2012.
- [5] Markus Berg, Bernhard Thalheim and Antje Düsterhöft: Dialog Acts from the Processing Perspective in Task Oriented Dialog Systems. In: Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2011), pages 176-177, Los Angeles (USA), 2011.
- [6] Benjamin Engber : 'How to Compare NoSQL Databases: Determining True Performance and Recoverability Metrics For Real-World Use Cases' - NoSQL matters ( <http://2013.nosql-matters.org/cgn/abstracts/> ), KÄln, Germany, 2013
- [7] 'Ähnlichkeitssuche in Multimedia-Datenbanken - Retrieval, Suchalgorithmen und Anfragebehandlung' - Ingo Schmitt, Oldenbourg Verlag
- [8] Inside SAP HANA - Inside SAP HANA - optimising data load performance and tuning - by John Appleby
- [9] SAP HANA Developer Guide - SAP HANA Appliance Software SPS 05, Document Version: 1.2 - 2013-05-14
- [10] Free ebooks - Project Gutenberg ( <http://www.gutenberg.org/> )
- [11] Twitter developer/API support site - <https://dev.twitter.com/>
- [12] IMDM 2013 1st International Workshop on In-Memory Data Management and Analytics - <http://www-db.in.tum.de/other/imdm2013/program.html>
- [13] No-SQL conference Cologne - <http://2013.nosql-matters.org/cgn/abstracts/>
- [14] Jiri Schindler: I/O Characteristics of NoSQL Databases. Tutorial at VLDB conference, August 2012, Istanbul
- [15] Text retrieval conference - <http://trec.nist.gov/proceedings/proceedings.html>
- [16] Springerlink system demonstration - ( [www.youtube.com/watch?v=q\\_2LdO-TOhM](http://www.youtube.com/watch?v=q_2LdO-TOhM) )

# Abstraction metaphors: a unifying view of modeling

Roland Kaschek

Searching for a job  
rolandkaschek@gmail.com

**Abstract.** Metaphor has been identified as a key tool of speakers of the English language. In conceptual modeling frequently English language sentences are created. Metaphors therefore should be well-known that structure conceptual models. This is, however, not the case so far. In this paper I point out a small number of metaphors that indeed cover the major abstraction mechanisms of conceptual modeling. These metaphors provide a non-technical unifying approach to conceptual modeling. They therefore may be helpful to learners and teachers of computer science or programming.

## 1 Introduction

The teaching of computing often rests on two things. The syntax of key constructs of selected modeling or programming languages is explained and not necessarily after that these constructs are illustrated by example. The related learners are then supposed to find their own kind of unifying understanding of all that. This might not be the most effective approach possible. A simple unifying, non-technical approach that can explain most of the things, the learners need to master, could be outlined by the teachers. This could boost the learning process. In this paper I am going to present an approach for that. My approach rests on two things, namely metaphor and abstraction. Abstraction is commonly considered key for modeling and metaphor is a key tool of speakers of English. It is therefore clear that both tools are going to be used by learners of computing anyway. In this paper I just suggest to use them in a planned and orchestrated way to aid learners in mastering their subject.

To understand a given universe of discourse requires to understand that which, from one's perspective, is invariant and that which is in flux. Models in computing therefore inevitably target invariance or change. Each of these two concepts can be defined in terms of the other. Thus, once a choice is made for that which is invariant or changing then the other term is defined by negation. Commonly both, change and invariance, in conceptual modeling are described by schemata. Schemata are meta language expressions that represent the elements of a set of object language expressions. In computing these object language expressions are claims about the state of affairs of their universe or discourse. Schemas for the description of invariance are usually used to specify

data or structure. Schemas for the description of change are usually used to specify activity sequences as used in imperative programming languages or business process modeling.

I am going to discuss abstraction in both data and processes. With regard to data abstractions are commonly discussed. This seems not to be so with regard to processes. I am going to show, however, that the same abstraction mechanisms that are used in data modeling also can be used in process modeling. As an archetypical example of process modeling I consider the use of imperative programming languages. Obviously I consider programming as modeling with executable languages. At first it seems unlikely that process modeling can involve abstraction since the process models only include instructions and do not focus on the actual instantiated process. However, after focusing on the cohesion between the process model's execution and the process stage the problem can be solved easily.

In the sequel I first discuss metaphor, turn then to abstraction and conclude with a discussion of abstraction in data and process modeling.

## 2 Metaphor

One of the metaphors Lakoff & Johnson [7] use to illustrate that metaphor pervades the whole language is "life is: a journey". They show that speakers of English often talk about human life as if it were a journey. The advantage of this practice is that someone, who knows what a journey is, already knows something about human life. For example, like a journey the life has a begin and an end. In between the two it may have a number of important or critical situations that structure one's thoughts and perceptions, i.e., the life may have stages that follow one after another. There might be such stages where one has to decide upon an important thing and choose between two alternatives, i.e., choose this or that path. A partner may come into someone's life and disappear after a while, i.e., join one's path for some time and then go their own ways again. There are more things that one can learn about our life if one talks about it as if it were a journey. Of course further metaphors can be used that provide insight in human life. Not necessarily, of course, these metaphors have to be logically coherent or consistent.

In the sequel I am going to use the form "unknown concept is: (a) known concept" to specify the metaphors I want to use. In the metaphor "life is: a journey" the concept "life" is the unknown concept and "a journey" is the known concept. Of course a metaphor can only aid in the knowledge transfer intended by its user if their interlocutor in fact knows or understands the "known concept" and finds it significantly easier to grasp than the "unknown concept". The concept of metaphor can be explained by a metaphor. For example, we can use the metaphor "metaphor is: stretching a term's area of application".

Metaphor has been used in computing to illustrate a number of different concepts[5]. For example, computers may be made more understandable by the knowledge transfer that results from the use of a suitable metaphor. As is well-

known the metaphor “computer is: a desk top” and “computer is: an interlocutor” are in use. Two metaphors that perhaps had even more far reaching impact on computing as the ones mentioned right now are ”function evaluation is: value construction” and ”function evaluation is: value lookup”. Obviously the former metaphor is cultivated in software engineering while the latter is cultivated in data engineering. In this paper I use metaphors to aid learners to acquire an understanding of data models and process models.

Abstraction is often considered as key to both, modeling and computing [1, 8, 4]. Therefore I focus in this paper on metaphors for abstraction, i.e., for different ways to abstract. It is, however, important to understand that modeling is not just about abstraction. Rather, among all those models that somehow incorporate the required abstraction usually modelers choose one that does so in a rather favorable way, given the case at hand. Of course what is considered as favourable not necessarily has anything to do with the problem to be solved. Rather the one who solves it and the circumstances of the solution might be the driving factors. In [4] I have called the distinguishing features between equivalent abstractions the problem representation and shown that representation is key to modeling rather than abstraction.

A metaphor that explains modeling is “modeling is: utilizing substitute problems”. It indicates that in modeling, rather than tackling a given problem immediately, one looks out for a substitute problem that can be solved instead and in some way or another is easier or better to solve and whose solution can be propagated back to the original problem and hence generates an acceptable solution of it.

Modelers experience a dichotomy with regard to choosing the substitute problem to be solved. There is the case of a formal narrative  $F$  that provides the context for the original problem  $P_1$  to be solved. Because of the formally specified context of  $P_1$  a substitute problem  $P_2$  might exist such that its solution  $S_2$  defines a solution candidate  $S_1^*$  of  $P_1$  that, possibly after adaption, can be proven to be a solution  $S_1$  of  $P_1$ . There is, however, also the case of the non-formal narrative  $N$  of the problem  $P_1$  and no proof might exist for the adapted solution candidate to be a solution of the original problem. In this case a verification is going to be necessary for the adapted solution of the substitute problem to be a solution of the original problem. Modeling always takes the flair of this case when the original problem is a problem within the real world.

### 3 Abstraction

A far reaching peculiarity of human language is its symbolic nature. This symbolic nature implies our ability to talk about things that are not physically present or to talk about things in imperfect ways. This allows us to lie, to leave out things, to talk about things in the past or the future and about the past, the future and the possible. Our language thus enables us to communicate and share with each other abstractions. One of our cognitive abilities likewise is fundamental for our capability to abstract, namely our ability to forget things. The

ability to forget things might, at first, be considered as a flaw or limitation of our cognitive apparatus. It is, however, necessary for a finite device to forget things it experiences [10] since otherwise some sort of information overload is the likely consequence. The abstractions I focus on are used within conceptual models as in use for software development. In this paper I neither attempt a general definition of abstraction nor of abstraction relationship. Rather, I only discuss specific abstraction mechanisms and their use for modeling invariance or change. A simple introduction into abstraction is in [6]. A sophisticated and also more complicated one is in [8]. In this paper I abstain from attempts to fully formalize my reasoning on abstraction. I rather focus on the abstraction mechanisms commonly used. The advantage of that is that no heavy formal apparatus is needed that would provide a coherent view on the abstraction mechanisms I discuss. The disadvantage of this approach is that questions are raised concerning the expressiveness of the various abstraction mechanisms considered. To some extent, however, I provide related answers.

Abstraction can be considered as a speech-act. It then means that a speaker  $\sigma$  lines out to their audience  $\alpha$  two symbolic objects  $I$  and  $A$  and declares that the latter is an abstraction of the former and that the abstraction mechanism capturing the relation between  $I$  and  $A$  is  $\mu$ . Abstraction therefore can be described as a five-place predicate  $\pi(I, A, \sigma, \alpha, \mu)$ . When it can be assumed that speaker and audience are known then within a model they do not have to be specified explicitly. It, moreover, in many cases may be assumed that speakers and their audience share a set of abstraction mechanisms and refer to them by conventional names. In conceptual modeling it is frequently assumed that the item  $I$  and its abstraction  $A$  are of the same kind. For example, in Entity-Relationship modeling the kind of item that may occur in a model is “entity type”, “relationship type”, “role” or “value set”. That each item and any of its abstractions are of the same kind obviously is a simplifying and restricting convention. If explicit coercion mechanisms are included into the modeling language then the said limitation is not a restriction of expressivity. I therefore suppose such coercion mechanisms to be available and admit only type-safe abstractions. According to [11] the adjective “abstract” dates back to the late 14th. century and originally was a term used in the grammar of nouns. It derives “from Latin *abstractus* ‘drawn away,’ past participle of *abstrahere* ‘to drag away, detach, pull away, divert;’ also figuratively, from *ab(s)-* ‘away’ (see *ab-*) + *trahere* ‘draw’ (see *tract* (n.1))”. I consider an abstraction relationship in a domain  $D$  therefore as a relation  $\rho$  among the entities of that domain that is asymmetric, i.e.,  $(x, y) \in \rho$  and  $x \neq y$  imply  $(y, x) \notin \rho$ ; reflexive, i.e.,  $(x, x) \in \rho$ ; and transitive, i.e.,  $(x, y), (y, z) \in \rho$  implies  $(x, z) \in \rho$ . Such relation trivially has no directed cycles. Rooted trees are an archetypical example of such relations.

The authors of [1, 8] point out that in the literature a view is prevalent that effectively conceives abstraction as a mapping from one modeling language to another. According to the general homomorphism theorem, however, each mapping  $f$  splits into the composition of an injection  $\iota$  and a surjection  $\sigma$ , i.e., such that  $f = \iota \circ \sigma$ . Now the injection  $\iota$  actually only achieves a renaming and



so I do not consider it as an abstraction and consequently do not agree with the view from the literature. I provide a more profound discussion of the metaphor “abstraction is: mapping” after I have introduced the abstraction metaphors “abstraction is:feature neglect” that I consider more fundamental. I am aware of two additional likewise fundamental abstraction metaphors, namely “abstraction is: implication” and “abstraction is: container neglect”. These metaphors likewise explain fundamental ways to abstract.

When one operationalizes the metaphor “abstraction is: feature neglect” then one considers an entity  $E$  and a list  $F$  of its features and a subset  $F' \subseteq F$  of them. One goes then on to consider entities  $A = (E, F')$  and  $I = (E, F)$  and says that  $A$  is an abstraction of  $I$ . Clearly feature neglect abstraction is reflexive, asymmetric and transitive. Consider, as an example, the concept  $I$  of human individuals that can be described by the attributes name, date of birth, gender, nationality, domicile, profession, occupation and title. Then one can talk about an abstraction  $A$  of  $I$  that does not have the gender, date of birth and nationality but still has the other features of  $I$ . Clearly this fits the said format. To see this take for example the entity  $H$  of human individuals in general and consider  $I = (H, \{f_1, f_2, f_3, \dots, f_8\})$  and  $A = (H, \{f_1, f_2, f_3, \dots, f_5\})$ , with the required interpretation of the features  $f_1, f_2, f_3, \dots, f_8$ . Then clearly  $A$  is a feature neglect abstraction of  $I$ . This form of abstraction is very often connected to the so-called generalization. If an entity is conceptualized as a composite entity, such as a book, that might be considered as an aggregate of chapters, then the feature neglect may take either of at least two forms. These are vertical and horizontal neglect. In vertical neglect features of the book would be ignored in such way that it would be conceived as a generalization of a book. In the horizontal neglect one would focus on parts of that composite structure of the book and ignore the rest. Not necessarily but quite often this would result in focussing on an aspect of a book that would not result in a generalization of a book. For example, one might want to ignore everything other than the fonts that were used in the book to set its body.

To demonstrate how basic the “abstraction is: feature neglect” metaphor is I show that generalization, classification and aggregation can be considered as instances of it. Essentially I have already (see the example above) done so with regard to generalization. I consider next the case of classification. Classification means to group the elements of a set  $S$  into a set  $\{S_i \mid i \in I\}$  of disjoint subsets  $S_i \subseteq S$  of  $S$  whose union is  $S$ , i.e., such that  $S = \cup_{i \in I} S_i$ . Such grouping is called a decomposition. For each  $s \in S$  we can introduce its feature list  $F_s = \{s' \in S \mid s \neq s'\}$  and understand each feature  $s'$  of  $s$  as an element of  $S$  that is considered as different from  $s$ . Then obviously the two sets  $S$  and  $\mathfrak{S} = \{(s, F_s) \mid s \in S\}$  mutually define each other. So instead of  $S$  we can consider  $\mathfrak{S}$ . To the latter, however, feature neglect can be applied and for any given relation  $\rho \subseteq S \times S$  we can define the the factor set  $S/\rho$  induced by  $\rho$  by feature neglect. In fact we can consider the set  $\mathfrak{S}_\rho = \{(s, F_{\rho,s}) \mid s \in S\}$  where we define the feature list  $F_{\rho,s}$  of  $s$  as the set  $F_s \setminus \rho(s)$ . Here  $\rho(s)$  is just the equivalence class of  $s$ , i.e., the set of elements of  $S$  that under  $\rho$  are considered equal to  $s$ . This shows that

classification can be considered as a case of feature neglect. For sake of purity I have not used the naming abstraction to introduce names for the equivalence classes after  $\rho$ . Aggregation can be modeled similarly. The only difference to the classification case is that the relation  $\rho$  does not specify the elements that are considered equal. It rather specifies those elements that occupy the same role, i.e., with regard to the aggregate functionally behave the same.

In the literature, as I read Henderson-Sellers, one has considered the metaphor “abstraction is: mapping” because in the target language of the abstraction the operations at ones disposal may be different from those at hand within the source language. However, even this can be modeled by feature neglect. Suppose that the union of such operations in the source and the target language is  $P$ . Suppose, moreover, that for each  $s$  in the source language the sets  $A_s, D_s \subseteq P$  are the sets of elements of  $P$  that are ascribed and denied to  $s$ , respectively. Then the feature list  $F_s$  can be extended by  $A_s$  and  $D_s$  and the likewise extended feature list  $F_{\rho,s}$ , by neglect of features, can now be used to specify which of the operations in  $P$  are supposed to be applicable to those elements of  $S$  that are considered equal under  $\rho$ .

When one uses the metaphor “abstraction is: container neglect” then one considers physically or only conceptually existing nested containers of items. The metaphor says that it is a form of abstraction to remove an item  $x$  from a container  $C$  that happens to be inside another container  $C'$  and to put  $x$  into  $C'$  instead. It is obvious that this metaphor easily can be reconciled with the “abstraction is: feature neglect” metaphor if one would consider the feature list of each item within the nested containers as being comprised of the containers an item is contained in transitively. Then removing an item from a contained container and putting it into one of the containing containers can be expressed as a case of the “abstraction is: feature neglect” metaphor and vice versa. In terms of abstraction operations that can be expressed with these two understandings of abstraction there is thus no difference. It may, however, not be particularly desirable to explicitly maintain the feature list in case abstraction is understood as container neglect. That might for example be the case if the collection of containers is changing during discourse that has generated the need for that abstraction.

When one uses the metaphor “abstraction is: implication” then one considers sets of assertions  $I$  and  $A$ . One says that  $A$  is an abstraction of  $I$  if the latter implies the former. For example, if this metaphor is presupposed to be true then the assertion “James is a band leader” is an abstraction of the assertion “James is a successful band leader” because the assertion “James is a successful band leader” implies the assertion “James is a band leader”. Clearly “James is a successful band leader” can be understood as the set {“*James is successful*”, “*James is a band leader*”} and since the set {“*James is a band leader*”} is a subset of the former set the former implies the latter set.

The metaphors “abstraction is: feature neglect” and “abstraction is: implication” do not share the domain of definition. The former is about concepts that can be specified by means of feature lists and the latter is about assertions,

i.e., utterances that either are true or false. To compare the expressiveness of these two metaphors first a way needs to be found to translate them into each other. For that purpose we consider now a universe of discourse  $U$ <sup>1</sup>. Abstraction between any two of the elements of a relational structure  $U_i$  of  $U$  can be understood as abstraction between two concepts if concepts are conceived as references to these elements. Therefore, to relate the two metaphors to each other we restrict to concepts that can be defined by a feature list where each feature is a true assertion about the relational structure  $U_i$  associated to the current state  $i$  of  $U$ . Suppose there are two such concepts  $I$  and  $A$  and let  $A$  be an abstraction of  $I$ . Then there exists an entity  $E$  in  $U_i$  with feature sets  $F_I$  and  $F_A$  such that  $I = (E, F_I)$  and  $A = (E, F_A)$ . Since  $A$  is an abstraction of  $I$  it follows that  $F_A \subseteq F_I$ . Consequently  $A$  is an implication abstraction of  $I$ . This means that the “abstraction is: implication” metaphor is at least as expressive as the metaphor “abstraction is: feature neglect”.

Call the universe  $U$  crisp for a state  $i$  if the set of assertions  $\mathfrak{X}$  about  $U_i$  is orthogonal, i.e., that  $X = X_j$ , for some  $j \in \{1, 2, 3, \dots, m\}$ , whenever  $\bigwedge_{k \in \{1, 2, 3, \dots, m\}} X_k$  implies  $X$  and  $X, X_1, X_2, X_3, \dots, X_m \in \mathfrak{X}$ . Let now  $I$  and  $A$  be sets of assertions in  $\mathfrak{X}$  such that  $I$  implies  $A$ . Then  $\bigwedge_{j \in I} j$  implies  $\bigwedge_{a \in A} a$ . Consequently, for each  $a \in A$  also  $\bigwedge_{j \in I} j$  implies  $a$ . Since  $U$  is crisp for  $i$  for each  $a \in A$  a  $j \in I$  exists such that  $j = a$ . Thus  $A \subseteq I$ . Consider now  $I$  and  $A$  as features of  $U_i$ . Then  $E_A = (U_i, A)$  is a feature neglect abstraction of  $E_I = (U_i, I)$ . Here only under the assumption that universe be crisp at its state  $i$  it was shown that feature neglect abstraction is a case of implication abstraction. This could mean that the latter is more expressive than the former.

We can, however, also consider the relationship between these two ways of abstraction in a different way. Let for some state  $i$  of the universe  $U$  the relational structure  $U_i$  be such that each predicate  $\pi$  about it is characterized by the models of  $\pi$  in  $U_i$ . Then any sets  $A$  and  $I$  of assertions about  $U_i$  can be characterized by their models  $\mathfrak{M}_A$  and  $\mathfrak{M}_I$ , respectively, in  $U_i$ . We thus can suppose that  $A$  is characterized by  $(U_i, \mathfrak{M}_A)$  and  $I$  is characterized by  $(U_i, \mathfrak{M}_I)$ . Let us denote the non-models of  $A$  and  $I$  among the structures in  $U_i$  by  $\mathfrak{N}_A$  and  $\mathfrak{N}_I$ , respectively. Then we can suppose that  $A = (U_i, \mathfrak{N}_A)$  and  $I = (U_i, \mathfrak{N}_I)$ . Let now  $A$  be an implication abstraction of  $I$ , i.e., let  $I$  imply  $A$ . Then each model of  $I$  is also a model of  $A$  [3]. Consequently the non-models of  $A$  are not models of  $I$ . This implies that the non-models of  $A$  are non-models of  $I$ . But this means that  $A = (U_i, \mathfrak{N}_A)$  is a feature neglect abstraction of  $I = (U_i, \mathfrak{N}_I)$ . It would seem that for sufficiently large universes the presupposition is true that each predicate about its current instance  $U_i$  is characterized by its set of models. If that would be true then feature abstraction and implication abstraction could be considered of the same expressivity.

<sup>1</sup> A universe of discourse  $U$  is a family  $U = \{U_i\}_{i \in I}$  of relational structures  $U_i$ , i.e., each  $U_i$  is a set for which a set of subsets  $E_i \subseteq U_i$  as well as a number of subsets of Cartesian products of these subsets is defined. Often the set  $I$  is supposed to be a linear order. Finally each index  $i$  is called a state of  $U$ .

A rooted tree  $R = (r, \rho)$  on a set  $D$  of items is a relation  $\rho \subseteq D \times D$  such that there is exactly one path from  $r$ , the root of the rooted tree, to each other vertex  $x \in D$  of the rooted tree. Each rooted tree can be considered as an instance of feature neglect abstraction. To see this consider a vertex  $x \in D$  of the tree. Then there is exactly one path  $P_x = e_1, e_2, e_3, \dots, e_{m_x}$  from  $r$  to  $x$ . The only path from any vertex  $x$  to itself is the empty path. Now each vertex  $x$  of  $R$  can be defined to be the pair  $(r, P_x)$ . Consequently each vertex  $y$  between  $r$  and a vertex  $x$  is a feature neglect abstraction of  $x$ . Feature neglect abstraction, however, is a little more general than rooted trees as it allows directed acyclic graphs to appear. Consider for example a domain  $D = \{r, x, y, z\}$  in which  $z$  has exactly the features  $a$  and  $b$  and  $x$  and  $y$  have exactly the features  $b$  and  $a$ , respectively. Then both,  $x$  and  $y$  are feature neglect abstractions of  $z$ . Moreover  $r$  is a feature neglect abstraction of  $x$  and of  $y$ . There are thus two paths between  $r$  and  $z$ . This turns  $D$  with the feature neglect abstraction into a directed acyclic graph that is not a rooted tree.

## 4 Abstraction in data modeling

As far as I know the most important abstraction mechanisms between any two items in a data model are instantiation, generalization, aggregation (aka part-whole relationship) and naming. Generalization and aggregation as early as 1977 have already been discussed by Smith & Smith [8][p. 79]. I am going to briefly discuss them here. A typical use of the instantiation can be illustrated by considering the metaphor “Unicorn is: a fictitious character”. It explains the term unicorn, rather vaguely as I admit, by the term fictitious character. This metaphor’s basic assumption is that an interlocutor will know about fictitious characters but not about the unicorn. The metaphor considers the unicorn and states that the unicorn is a character whose existence is not of the real world. It says that the Unicorn exists in fiction and related narration only. Considered extensionally the instantiation “Unicorn ISA fictitious character” specifies that the Unicorn is included in the set of fictitious characters that is under consideration in the model that contains the said abstraction relationship.

A typical use of the generalization can be briefly illustrated by considering first the metaphor “German shepherd dog is: a dog”. It explains the term German shepherd dog in terms of the term dog and presupposes that the latter is known by an interlocutor while the former is not. The metaphor considers the race German shepherd dog and says that each instance of it in fact is an instance of the kind dog, i.e., that it is a dog. Since each German shepherd dog is a dog it must have each of the features of a dog. Since, however, there are dogs who are not German shepherd dogs the German shepherd dogs must have features that are not shared by at least one other race of dogs. The extensional view of the metaphor thus can be translated into an intensional one. Namely, the race German shepherd dog can be obtained from the kind dog by adding the features typical for German shepherd dogs. Of course, also vice versa, the kind dog can be obtained from the race German shepherd dog by removing the features not

typical for the kind dog. At the first glance it might be a little confusing that entities with many features are element in a set that only has only a few of these features. In particular this might be so, if the dogs in question would be represented by tuples, since, for example for natural numbers  $m, n$  with  $m \neq n$ , an  $m$ -tuple is not an  $n$ -tuple. However, if each dog is represented by a partial function then the problem goes away. Obviously each partial function that is defined on all the features of the race German shepherd dog also is defined on all the features of the kind dog.

Aggregation is about composing entities out of other entities. It has already been shown above to be a special case of feature neglect abstraction. It is a means of describing how a higher level entity, the aggregate, is created out of lower level entities, the constituents. That creation may rest on physical attachment or on some kind of coordination. A number of different kinds of aggregation can be explained by metaphors: “aggregation is: attachment”, “aggregation is: communication” and “aggregation is: rule sharing”. The metaphor “aggregation is: attachment” can be explained with a simple example such as a motorcycle. A motorcycle, in the traditional construction, consists of parts such as a frame, a motor, a gear box, the break system, the illumination system, the fuel system, the wheels, the propulsion system and its steering. Each of the other motorcycle parts in some way is physically attached to the frame that has no other vital function. The metaphor “aggregation is: communication” can be illustrated by a dinner party. The party guests create a group of people, mainly, by the conversation they have. That conversation may lead to the guests make certain common decisions such as to raise their glass and have a toast or similar. The metaphor “aggregation is: rule sharing” can be explained by the birds in a flock, the fish in a school, the trucks or ships in a convoy, or similar. As far as I know it is unknown today how the birds in a flock manage to attain and keep the energy efficient flight formation. Most likely they use the same behavioral rules plus observe what their neighbours are doing and how the air flows. Following common rules, however, seems to be an important issue here since it would make superfluous a communication of a larger number of individuals. An interesting version of the “aggregation is: rule sharing” metaphor appears in Conway’s game of life. The rules that apply to each cell can even bring into existence aggregates whose constituents are replaced completely very rapidly.

The metaphor that remains to discuss is “abstraction is: naming”. Please note that this has nothing in common with the renaming that was discussed above. Renaming was about using a new name instead of an old one. Naming is about introducing a name at all. A name for an entity that one can refer to is a short-hand substitute of that schema of reference used to refer to the entity. Naming, first of all, is fundamental for all kinds of software development artifacts since within these artifacts names are used. The related definitions, i.e., reference schemas, are provided in the thesaurus (aka data dictionary). In Entity-Relationship modeling naming thus appears in the names of the entity-types, relationship-types and value-types the ER-diagram contains. A more subtle use of naming is the use of the role concept. For example, the relationship “hates”

between the girls in a high school class will involve two roles, namely “hater” and “hatee” in the obvious meaning. Suppose that Mary attends the class under scrutiny. She might for example be a hater of Josy, Pam and Mia. That turns these girls into hatees of Mary. Not necessarily any of these girls is a hater of Mary. Roles, by the way, may be composed into path expressions. As is well-known roles banning roles from ER-modeling would reduce its expressiveness.

The definition of an entity or an entity set in a given universe of discourse is a sequence of instructions about how to refer to that entity or entity set. If each of these instructions is considered as a feature of that entity or entity set then the name of it can be introduced as a renaming of the feature neglect abstraction that gets rid of all these features.

## 5 Abstraction in process modeling

Processes commonly are conceived as sequence of process stages. Imperative process modeling does not focus on these stages, however. Instead the operations are modeled that create them. That blurs a little the way towards understanding abstraction as key tool of process modeling. One can, however, consider the process stages as abstractions of the execution of the mentioned operations. The operations commonly used in imperative programming are assignment, alternative, loop and sequence. I am going to show below that feature neglect abstraction is sufficient to understand how these operations determine the process stages.

In his famous paper [2] Tony Hoare has defined a formal semantics for these operations. Based on it in principle it is possible to prove the correctness of an imperative program against its specification as a function. I am not aiming to contribute to the related theory. My aim is much less ambitious. I just point out the similarity of process modeling (with imperative executable languages) to data modeling. Let for this end be a process specification  $P$  be given as a sequence  $P = \{O_1, O_2, O_3, \dots, O_m\}$ . Then for each index  $i$  the process stage  $S_i$  is the list of pairs  $(n, v)$  of all the variables  $n$  whose value  $v$  is affected by  $O_1, O_2, O_3, \dots, O_i$ . The operation  $O_{i-1}$  transforms  $S_{i-1}$  into  $S_i$  and  $S_i$  is an abstraction of the execution of  $O_{i-1}$ . As I did before I discuss these abstraction relationships based on suitable metaphors.

To explain the assignment operator I use the “assignment is: named term neglect” metaphor. To see how it matches the structure of the discussion of assignment consider an assignment operation  $a := \phi$ , where  $a$  is a variable and  $\phi$  a term. The variable and the term are considered as the assignment’s target and source, respectively. A variable and a term is determined by its name and value and its formula and the values of the variables in the formula, respectively. The formula is a modeling language construct and upon evaluation provides a values. The evaluation of the formula can be described by a tree. The assignment operator, however, upon evaluation of the term, discards that tree and only keeps that value which then is then associated with the target’s name. The effect of the assignment is therefore obtained by the composition of a naming and a neglect abstraction.

If the process state prior to executing the assignment is  $S$ , then the execution of the assignment can be specified as a pair  $(S, \{o_1, o_2, o_3, \dots, o_m\})$ , where the  $o_i$  are the outcomes of the individual steps  $s_i$  in the evaluation of the formula's tree. What is retained after execution of the assignment, if side-effects are excluded, is just  $(S, \{o_m\})$  which obviously is a feature neglect abstraction of  $(S, \{o_1, o_2, o_3, \dots, o_m\})$ .

Consider for example the function

$$y = \begin{cases} -2x & \text{if } x > 0 \\ 1 & \text{if } x = 0 \\ 2x & \text{if } x < 0 \end{cases} .$$

It can be implemented as an assignment like this:

$$y := -2\text{sign}(x) \cdot x + 1 - \text{sign}(\text{abs}(x)).$$

In this assignment on the right hand side of the assignment operator there is the formula and upon its evaluation the value appropriate for the function will be computed. It is not difficult to conceive a sequence of steps to evaluate that term. This abstraction also is important because it permits to change the term to be evaluated while nothing else in the model needs to be changed. For example the more conventional choice could have been used to gain the same effect:

$$y := \text{if } x > 0 \text{ then } -2x \text{ elif } x = 0 \text{ then } 1 \text{ else } 2x.$$

The alternative can be explained using the metaphor “alternative is: path neglect”. In a simple form the alternative can be specified as

$$\text{if } c \text{ then } o_c \text{ elif } d \text{ then } o_d \text{ else } o.$$

It means that in case the predicate  $c$  is the true the operation  $o_c$  is executed and if  $d \wedge \neg c$  is true the operation  $o_d$  is executed and otherwise operation  $o$  is executed. One or more of these operations may actually be void, i.e., a nop operation. The said metaphor addresses that situation by focusing on the fact that in each of the mentioned cases the appropriate operation has been specified and thus will be executed. Therefore, which of the conditions was true, i.e., which of the offered paths was taken, does not matter. In imperative programming languages the alternative is also known as “if” or “case-statement”.

Let the process state prior to executing an alternative be  $S$ . Denote the verification of the predicate  $\pi$  by  $v_\pi$  and the execution of the operation  $o_\pi$  by  $e_{o_\pi}$ . If in the execution of the alternative the path will be followed that is associated to the predicate  $\pi$  then that execution can be described by the pair  $(S, \{v_\pi, e_{o_\pi}\})$ . Obviously  $(S, e_{o_\pi})$  is a feature neglect abstraction of it.

To define finer granular predicates in a systematic manner nesting of alternatives may be used. Consider for example the three variables  $a, b$  and  $c$ . Suppose that the simultaneous assignment style such as in Python is used, see e.g. [9]. That style simply avoids the use of additional variables and thus makes the code easier to understand. Consider the program  $P(a, b, c)$  in table 1:

```

def P(a,b,c):
  if a > b:
    if b > c:
      a, c := c, a
    elif a > c:
      a, b, c := b, c, a
    else:
      a, b := b, a
  elif b > c:
    if a > c:
      a, b, c := c, a, b
    else:
      b, c := c, b
  else:
    a, b, c := a, b, c

```

**Table 1.** The program  $P(a, b, c)$

Let it be such that  $P(a, b, c)$  transforms state  $S$  into state  $S'$ . Suppose that in state  $S$  it is unknown how the values of  $a, b$  and  $c$  are related to each other. The way the alternatives are nested here and equipped with operations entails that  $P(a, b, c)$  can be seen as a tree with six branches and that at the end of each branch the operation is performed that assures that post execution of  $P(a, b, c)$  it holds  $a < b < c$ . Thus the abstraction taking place consists in neglecting the path that was taken from the top of the tree to the leaf of it that ultimately is reached when  $P(a, b, c)$  terminates and assures that in  $S'$  it holds  $a < b < c$ .

The sequence of instructions  $I_1, I_2, I_3, \dots, I_m$  can be considered as a unit. It certainly can be understood as an aggregation complying to “aggregation is: comprehension”. As such it obviously is of the neglect kind of abstraction as for the aggregate it is neglected which instructions, how many of them and in which sequence appear. The only thing that is retained is the composition of functions on the data space that are associated to the instructions  $I_i$ . Each instruction, as suggested by Hoare, induces such function and the sequence of instructions just induces the composition of them.

The loop instructions can be explained by the metaphor “loop is: history neglect”. For any block  $B$ , i.e., composition of instructions, and predicate  $p$ , if it terminates the loop instruction  $L \equiv \text{while } p \text{ do } B$  defines a process stage  $S'$ . Suppose that prior to execution of  $L$  the process is in stage  $S$ . We are going to see that  $S'$  is a feature neglect abstraction of the execution of  $L$ . When the loop  $L$  is executed at all then there is a first, second, third,  $\dots$ , execution of the block  $B$ . Let for the  $i$ -th execution of  $B$  be  $c_i$  the set of pairs  $(n, v)$ , where  $n$  is a variable affected<sup>2</sup> by the loop’s execution so far and  $v$  is its value at the end of the  $i$ -th execution of  $B$ . The execution of  $L$  up to and including the  $m$ -th execution of  $B$  thus can be described by the pair  $(S, \{c_1, c_2, c_3, \dots, c_m\})$ . Suppose that  $L$  ends

<sup>2</sup> Here again I exclude the occurrence of side effects.



with this  $m$ -th execution of  $B$  then  $S'$  is characterized by the pair  $(S, \{c_m\})$  which is a feature neglect abstraction of  $(S, \{c_1, c_2, c_3, \dots, c_m\})$ . Neglecting the history of the loop execution means here to forget about  $\{c_1, c_2, c_3, \dots, c_{m-1}\}$ . Since this is so surprisingly simple I do not discuss a related example.

## References

1. Brian Henderson-Sellers. On the mathematics of modelling, metamodelling, ontologies and modelling language. Springer. 2012.
2. Tony Hoare. An axiomatic basis for computer programming. CACM (1969) vol.12, 10: pp. 576.
3. Wilfrid Hodges. Model theory. In: Edward N. Zalta (ed.), The Stanford encyclopedia of philosophy (Fall 2013) <http://plato.stanford.edu/archives/fall2013/entries/model-theory/>.
4. Roland Kaschek. A semantic analysis of shared references. Proceedings of ER 2013. Springer LNCS 8217: pp. 88.
5. Roland Kaschek, Alexei Tretjakov. Enabling metaphor evolution for improving systems' usability. JDIM (2006) vol. 4,4: pp. 243.
6. Roland Kaschek. A little theory of abstraction. Proceedings of Modellierung 2004. GI Edition Lecture Notes in Informatics, P 45. Bonn, Germany, 2004: pp. 75.
7. George Lakoff, Mark Johnsen. Metaphors we live by. The University of Chicago Press, 1980.
8. Lorenza Saitta, Jean-Daniel Zucker. Abstraction in artificial intelligence and complex systems. Springer. 2013.
9. Mark Lutz. Programming Python. Third release, O'Reilly. 2011.
10. Micheal O'Shea. The brain. A very short introduction. Oxford University Press, 2005.
11. <http://www.etymonline.com/index.php?term=abstract>, accessed Jan. 20th. 2014.

# Icon Recognition and Usability for Requirements Engineering

Sukanya KHANOM<sup>1</sup>, Anneli HEIMBÜRGER and Tommi KÄRKKÄINEN

*University of Jyväskylä, Department of Mathematical Information Technology, Finland*

**Abstract.** When we introduce icon-based language into the context of requirements engineering, we must take into account that what users perceive as recognizable and usable depends on their background. In this paper, we argue that it is not possible to provide a single set of visual notations that appeal to all of stakeholders. Instead, we suggest an adaptable preference framework, which generates personalized notations that correspond to personal background. We present and evaluate icon-based language: a new kind of approach to requirements engineering work to explore its possibility and usability. In an initial evaluation of students residing in Finland, results reveal that users are able to recognize a group of icons fairly well. Our findings show that an icon-based language could probably be a positive means in improving awareness of requirements engineering as it tends to take advantages of icons which are intuitively understandable to represent traditional textual requirements.

**Keywords.** Requirements engineering, icon-based language, stakeholders, culture, visual notation, experimental study

## Introduction

Requirements engineering (RE) is recognized as being one of the most difficult engineering tasks. Principally, RE has been categorized in terms of two areas: requirements development (RD) and requirements management (RM) [1,2]. Its merit for RD is primarily examined in this paper. Researchers have perceived that diversity of stakeholder's background can inhibit successful use of the RE process [3]. Although there is a growing body of research pertaining to requirements development and management across borders, information in the area of cross-cultural RE is lacking [3,4]. The tasks involved in RE are essentially collaborative. Various stakeholders participate in these tasks, and it is necessary to obtain all potential requirements from stakeholders. Requirements are commonly high identity risk, and they can be difficult to capture in situations where a communication gap exists between developers and users. Requirements identity depends on the background distance between relevant stakeholders [5]. In practice, it is more problematic for stakeholders to communicate requirements internationally than with local users. Particular types of communication problems that affect and limit shared understanding between developers and non-technical users include so-called ineffective communication channels, restricted notation languages, and cultural and organizational factors [6]. Poor communication can broadly be classified in three different ways: lack of articulation (the ability to

---

<sup>1</sup> Corresponding author.

express information), misunderstanding (the tendency for various stakeholders to interpret differently the same piece of information), and conflict (multiple perspectives and differences) [7,8].

There may be numerous approaches to solving the problems encountered in RE. This paper introduces one such approach with the help of the following key concepts: preference aspects, RE modelling and icon-based language.

*Preference aspects* assist in building a user preference framework with adaptive user interfaces that conform to the user's background.

*RE modelling* provides classes for representing the requirements element. It designates how requirements information is typically structured, as well as the relationship between different elements within a particular project. On a macro-level, we define elements within a project as requirement, attribute, traceability links and people involved in the project. *RE artefacts* are characterized as a static element, which means they are not dependent on preferences. Thus all users visualize requirements features or functions in the same manner.

*Icon-based language* is designed in visually supporting pieces of RE artefacts that go beyond ordinary textual description. Being dissimilar from RE artefacts that are static, icon-based information is dynamic, depending on the user's background. The divergence of icon presentations that adhere to preferences can be seen, for example, by illuminating icons to represent the information: users from Europe might prefer an interface with only icons while users in Asia may prefer an interface with icons supplemented by text captions.

Our research explores how well respondents are able to recognize a set of icons and how icon-based language can help to enrich RE work. To answer these two questions, this paper evaluates icons that represent RE attributes and surveys respondents' satisfaction. During the evaluation phase, there were two iterations: one with student participants and another with expert participants. The first iteration, presented in this paper, included 48 students in an RE course in the Department of Mathematical Information Technology at the University of Jyväskylä. A Web-based test was implemented in a MediaWiki environment. Our findings have the potential to show whether the use of icons is advantageous in the RE domain. We anticipate that the added functionality of icons could simplify RE tasks for a range of stakeholders.

In the following section, we introduce previous works on which our method for designing a concept of icon-based language has been based. In Section 2, we describe our research approach in terms of the three artefacts – *preference aspects*, *RE modelling* and *icon-based language* – that have been developed. In Section 3, we explain the implementation environment. In Section 4, we offer an empirical evaluation of icon-based information. In Section 5, we discuss our results and recommendations for improvement. In Section 6, we propose areas of future study and present our conclusions.

## 1. Related Work

Natural language description is a technique that aims at communication which is commonly understandable by all potential users. However, natural language has widely recognized limitations, such as potential ambiguity and inconsistency. Substituting natural language with more structured notations would probably decrease ambiguity [9]. Dozens of requirements engineering visual methodologies and techniques have been

developed and made available for respondents. Unfortunately, some of these lack mechanisms for managing complexity. In the absence of such mechanisms, problems have been represented as single massive diagrams, regardless of their complexity. For example, the ER model that has been used for several decades is still lacking such mechanisms. As a result, ridiculously complex diagrams are often produced in practices that overwhelm end users [10]. The goal-oriented model, a language which was developed more recently and specifically for communication with end users, also lacks these mechanisms [11]. It can be concluded that the lesson has still not been learnt after all this time. Otherwise, only a very small number of tools have been developed to support requirements modelling and requirements management. This suggests that there is room for improvement for those two perspectives [4]. Diagrammatic modelling techniques such as UML [12-14] are a popular approach amongst developers. Nonetheless, end users are at a disadvantage when asked to validate existing diagrams or notations, as they are typically required to translate their knowledge into an unfamiliar language [6,9]. Hence, modelling techniques may be unsatisfactory or even useless when communicating with non-technical users. A good requirements collaborative mechanism is comprised of several crucial influencers. Of these, one type of support is to be able to specify requirements using textual, graphical, and modelling descriptions (with rich visual aids such as images and icons) [9]. Another support feature is the ability to trace backward and forward between requirements [1,2].

As the RE paradigm has shifted from localization to globalization, developing software when team members are located in distributed geographic regions can pose many challenges for developers. A single technique for development of RE within a multicultural organization does not necessarily mean optimization. Likewise, the use of different methods and different uses of the same methods across countries can be problematic, depending on cultural differences. Asian websites, for example, tend to commonly be bright and colourful, with frequent animations that try to attract the user's attention. This degree of high complexity is often perceived as information overload by Westerners, who prefer more structured content. Asian people, in contrast, have been shown to efficiently filter such dense information [15-17]. This fact motivates researchers and developers to realize that adapting the interface to a user's culture is a major strategy to market success [18,19]. While cultural preferences have long been researched, efforts in this direction have been limited only to specific countries or regions [20-22]. The implementation of guidelines related to cultural preferences mostly emphasizes the design of the user interface or navigation system, which is only beneficial for users who belong to the group targeted by the company. Historically, O'Neill-Brown has proposed the idea of developing systems that can automatically recognize and adapt themselves to suit a user's cultural preference [22]. Later, Heimgärtner and Burgmann conducted research in the area of cultural adaptivity in navigation systems [20,21]. Since such adaptations are largely intended to improve the user's learning experience, they did not comprise a full range of user interface components. More recently, therefore, Reinecke [18,19] has suggested an increasingly adaptive approach that takes into consideration all interface components influencing a user's preferences, which vary according to their nation and culture.

Our contribution is novel in the way in which we present an alternative approach that applies icon-based language to represent the context of RD and that enables icon appearance to be adapted to the preferences of users of any culture.

## 2. Research Approach: Model of Requirements, Icon-based Language and Preference Dimensions

This research follows the research methodology of design science [23,24], which consists of three primary phases: identification of problems and objectives, design solution, and evaluation.

By scrutinizing a vast amount of literature in the first phase (e.g. [5,6,25]), we arrived at three research problems: (1) the difficulty for stakeholders to express their needs explicitly, (2) the difficulty for stakeholders to understand, communicate and review requirements, and (3) the difficulty for stakeholders to make requirements constant. To arrive at a potential solution to these problems, solid objectives of icon-based language must be defined. Three key benefits can be determined: (1) to assist stakeholders to specify and communicate requirements, (2) to support requirements analysts to prioritize and resolve conflicts, and (3) to enable RE stakeholders to investigate changes in requirements and to continue tracking the requirements life cycle.

In the second phase, the design solution stage, we extend three key insights of our approach from [26-29] that is, preference aspects, the requirements engineering context and icon-based language (Figure 1). The details of these three artefacts are explained in the next three sub-sections.

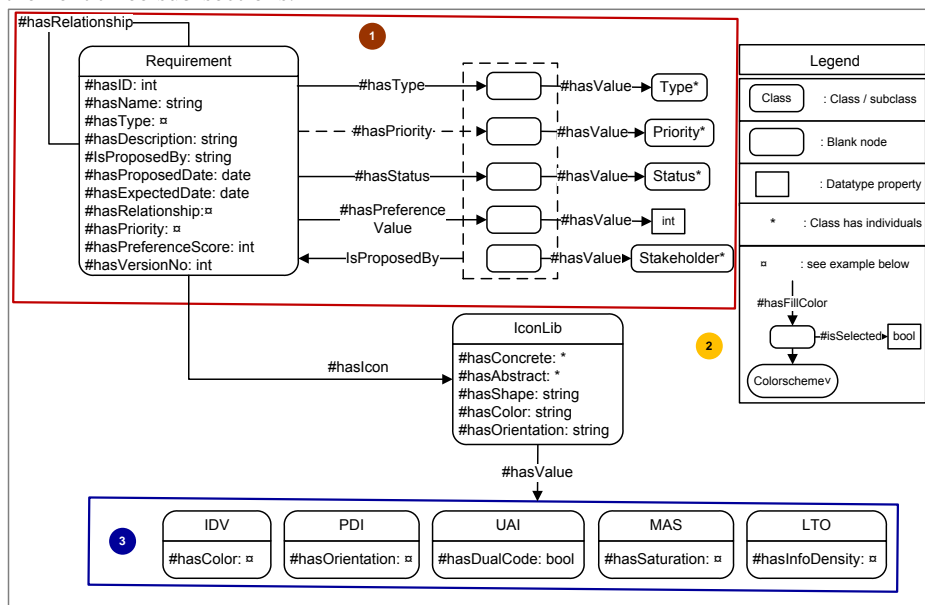


Figure 1. Model of requirements, icon-based language and preference dimensions.

### 2.1. Requirements Engineering

The central concept of requirement artefacts is correlation between attributes, stakeholders, and relationship (see number 1 in Figure 1). Attributes are the properties that distinguish a requirement from other requirements and establish a context and background for each requirement. Stakeholders refer to the persons or systems that have the purpose and ability to achieve goals. Relationships signify the interaction between two or more requirements. Each requirement is proposed by a stakeholder, and

thus it is essential to record information about associated stakeholders. We categorized the requirements into groupings (a taxonomy), in order to facilitate better organization and management. Eight types were created to delineate software quality and development process quality [30]. All requirements are characterized by a unique identifier (*#hasID*), name (*#hasName*), description (*#hasDescription*), type (*#hasType*), proposed stakeholder (*#IsProposedBy*), proposed date (*#hasProposedDate*), expected date (*#hasExpectedDate*), association (*#hasRelationship*), priority (*#hasPriority*), preference score (*#hasPreferenceScore*), and version number (*#hasVersionNo*).

## 2.2. Icon-based Language

It was first necessary to derive an icon library of visual notations designed to be attached to the requirement itself, the requirements process and the user interface (see number 2 in Figure 1). To promote scalability and variability, we built libraries for icon-based language to collect icons that relate to attributes that adhere to every requirement. Icons in this library must be designed in accordance with the cultural aspects of Hofstede's dimensions (see number 3 in Figure 1). Other icon syntactic properties are, for example, "position" (which characterizes the icon's orientation on the X and Y axes), "size" (which exemplifies icon size, including 1D iconic elements, such as lines, 2D iconic elements (areas), and 3D graphic elements (volumes)), "style" (which typifies colour and shape), and "link" (which symbolizes link attributes such as curves and dashed lines).

## 2.3. Preference Dimensions

We ground our preference framework in cooperation with cultural user ontology [19] whose conception was already validated, and the evaluation exposed that preference adaptive method has a competitive advantage over non-adapted version. Participants' preference is relied obviously on their personal background, namely education, gender, age and nationality.

Culture also plays an essential role in the use of information and communication technology. Cultures have different degrees of context: some cultures are determined as high-context while others are considered as low-context. In high-context communication, most of the meaning is found in the context. By contrast, in low-context communication, most of the meaning is in the transmitted message itself. Problems and conflicts frequently emerge when people from high- and low-context cultures communicate with each other [31-33]. In Table 1, we have summarized the rules for a cultural interface based on Hofstede's theory of the five dimensions of human-computer interaction components, such as colour, appearance and contents [34-38]. Power distance (PDI), for example, describes the extent to which hierarchies exist and are accepted by the members in a society. In countries that have been assigned a high power distance score, societal inequalities are much more acceptable than in low power distance countries. People in highly individualist (IDV) countries are usually seen as being more independent; in contrast, people in collectivist countries often see themselves as part of a group. The third dimension, masculinity (MAS), refers to a high preference for competitive achievement (high masculinity) versus low preference (femininity). The degree to which the members of society tolerate uncertainty and ambiguity is inversely reflected by an uncertainty avoidance index (UAI); that is, people from high uncertainty avoidance countries prefer less ambiguity than those in

low uncertainty avoidance countries. The fifth dimension, long-term orientation (LTO), measures how people perceive time. In LTO countries, people are comfortable sacrificing for long-term benefit, but in countries with short-term orientation people are more focused on immediate results. Table 1 illustrates adaptation rules for icon contents that represent an aspect of RE artefacts.

**Table 1.** Icon adaptation rules divided into high and low according to user preference frameworks.

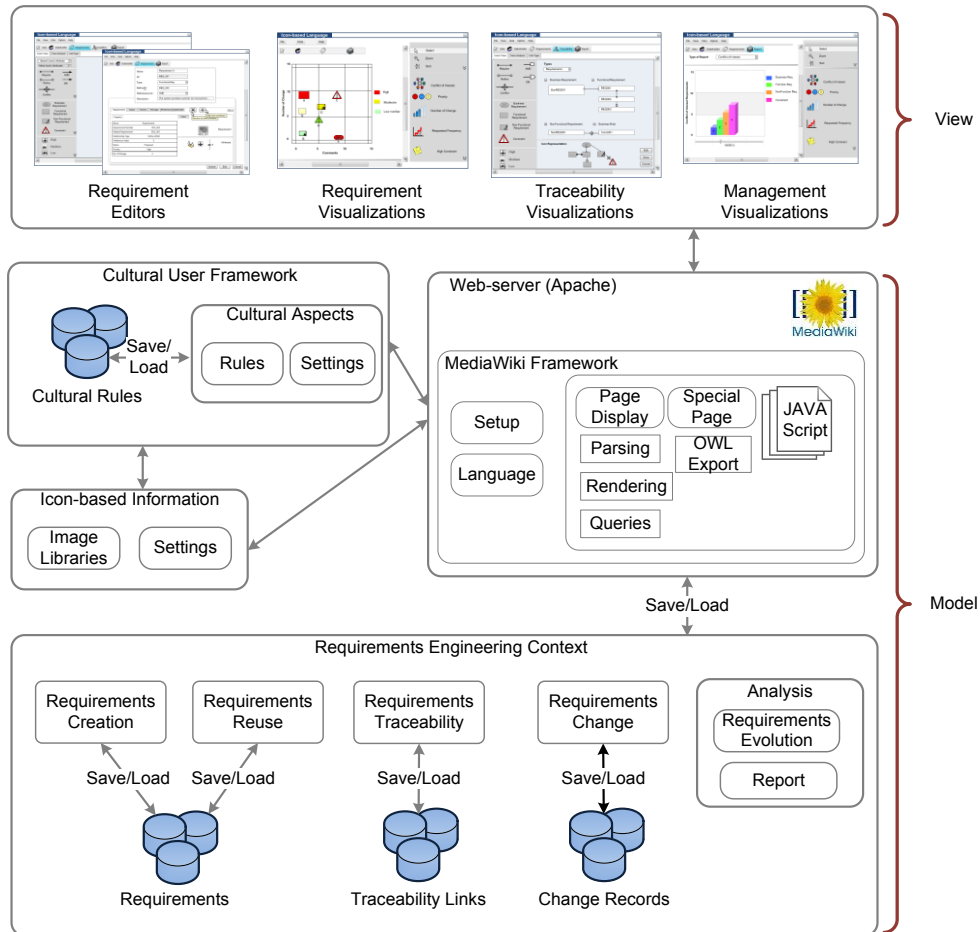
IL Aspect	Hofstede's Dimension	Dynamic Component (icon)	
		Low Scores	High Scores
Colour	IDV	Icons' display colours correspond to nation's preference.	Homogeneous colours.
Space/Orientation	PDI	The space between each icon element can be close to each other.	The space between each element must be wide and follow hierarchical structure.
Support	UAI	Icon individually shown without text caption.	Icon displayed with dual-coding (text and icon).
Information Density	LTO	Low number of icon details. Limited number of shapes.	High number of icon details. Many kinds of shapes.
Saturation	MAS	Icons' saturation corresponds to nation's preference.	Pastel-coloured icons.

### 3. Implementation in MediaWiki Environment

Wikis are defined as collaboratively and freely expandable collections of interlinked webpages that allow users to edit the content of a webpage [39]. There are a wide variety of wikis available, including MediaWiki. In order to develop a collaboration tool to communicate requirements, we extended the MediaWiki architecture. Wiki technology facilitates stakeholders to work on the same thread without overwriting each other's modifications, with the added benefit of being able to keep track of each other's contributions. The concept of maintaining multiple versions in a Wiki originated from similar mechanisms implemented in software version control systems [39]. Through wiki-based collaboration, many stakeholders are able to view the newest version, control or manage concurrent write-access, and implement rollback to prior versions. Once requirements have been completed, the page is released for others to see, review and further modify. The basic MediaWiki model does provide for collaboration and distribution, but does not provide support for RE (such as requirements creation, requirements reuse, and requirements evolution analysis). Furthermore, the basic model does not provide details of attributes and properties that should be stored within the model. It is up to developers to take advantage of those and to customize advance supportive features.

Figure 2 describes the technical implementation for an icon-based language prototype. The top level demonstrates the user interface view, which consists of four views: requirement editor, requirement visualization, traceability visualization and management visualization. One important trait of icon-based language is that it is comprised of four layers. In the top layer, a user interface that is kept separate from the system model allows for each tier to be easily modified with minimal impact on the others. The lower level of the framework illustrates the convergence of three main artefacts –user preference framework, icon-based information and RE artefacts – to reach an icon-based information adaptivity process. The icon-based interface is tailored to the user on the basis of adaptation rules (see Table 1). For instance, if a user has a

high score in UAI, then an interface with very simple, clear imagery and limited choices is provided.



**Figure 2.** A framework for icon-based language in the RE context for multi stakeholders.

#### 4. Experiments on Icon Recognition and Satisfaction

In order to evaluate the advantage of icon-based language, our experiment was aimed at measuring user recognition and satisfaction. We extended the result [40] by including additional three participants and by supplementing other two facet evaluations, that is, recognition and usability. We hypothesized that the benefit of icon-based language is twofold. First, we expected that icon-based language that uses icons to represent textual improves recognition (Hypothesis 1). Secondly, we assumed that icon-based language increases satisfaction (Hypothesis 2).



#### *4.1. Method*

##### *4.1.1. Participants*

In order to balance education level, participants had to be students. We invited 48 students attending a RE course at the University of Jyväskylä to participate in the survey.

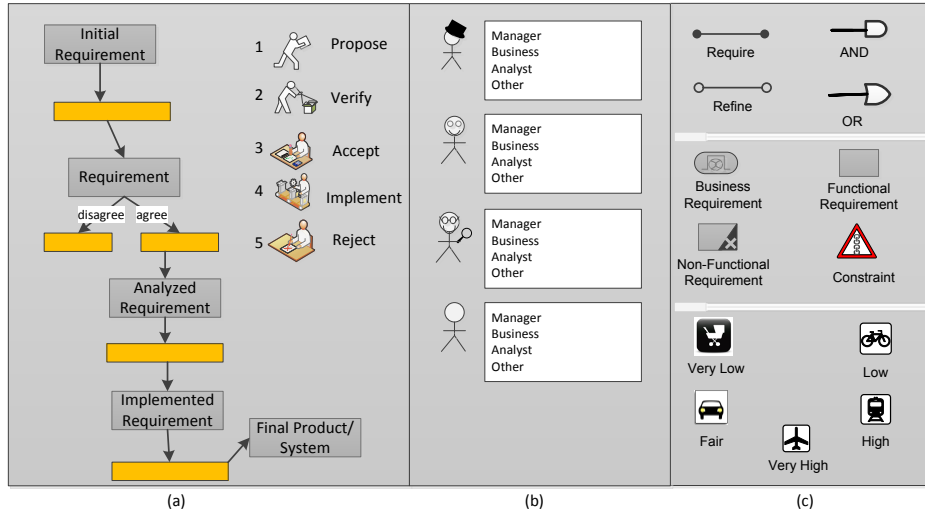
##### *4.1.2. Apparatus and Procedure*

We conducted the experiment on MediaWiki as a web-based survey. The icons were presented to participants on webpages. During the experiment, participants encountered three sections: background information, a form for personal information, and a test on icons. Each section contained a description that helped users to complete the tasks.

On arrival, participants received a verbal explanation about the test procedure, followed by a short questionnaire to solicit information about their background (nationality, education level and work experience). Frequency of computer usage, age and gender were not recorded. Participants were then provided with a short introduction to the purpose and functions of icon-based language, as well as an explanation of its structural categories. The explanation and the questionnaire were provided in English. The test procedure consisted of two subsets: icon recognition and a survey on satisfaction.

The first subset consisted of three tasks: individual icon recognition, multi-icon recognition and compound icon recognition. For individual icon recognition (see Figure 3(a)), participants encountered diagrams on a requirement's life cycle, consisting of five blanks, and a list of potential icons. They needed to select and place the appropriate icon into its corresponding diagram state. For multi-icon recognition (see Figure 3(b)), respondents were confronted with 14 icons of three types: five icons for priority type, five icons for status states and four icons for stakeholder type. All icons were abstract, so that their meaning must be guessed. Icons in each of the three types could be clearly inferred from their characteristics. First, various degrees of vehicles' velocities were utilized to represent how important and urgent a particular requirement is (e.g. the slowest vehicle, a baby carriage, signified very low priority). Secondly, the interrelation between a person and his or her environment was used to represent life cycle states from submitting to verifying (e.g. a person writing a red X sign was representative of a rejection status). Thirdly, different types of actors could be distinguished by various stick figures [11,13] (e.g. a manager was shown by a stick figure wearing a hat). Finally, for compound icon recognition (see Figure 3(c)), respondents were encouraged to construct a sentence from pre-defined icons. We mimicked the goal-oriented model for this test, but rather than using existing goal notations, we defined icons and shapes for our own icon-based language.

The second subset was composed of three questions, plus two freeform areas for opinions and comments. Participants were asked to proceed step by step, beginning with the first task of individual icon interpretation to the last part of the questionnaire.



**Figure 3.** The test tasks: (a) individual icon recognition exemplifying the requirement's life cycle, (b) multi-icon recognition typifying requirement attributes, and (c) compound icon recognition characterizing requirement types and relationships.

## 4.2. Results

### 4.2.1. Icon Recognition

To determine icon recognition, we based our evaluation on the frequency of prediction accuracy. For analysis, for each task and participant we coded a correct prediction (success) with a 1 and an incorrect answer (failure) with a 0. To test whether icon-based language reached a significantly higher frequency, we employed the statistical mechanism of binomial confidence intervals (with confidence level 0.95 and  $\alpha = 0.05$ ). Depending on the task, the participants correctly predicted the icons' meaning with a minimum of 25 and maximum of 41 items (mean = 34.67, sd = 10.12), with an overall prediction accuracy of 73 percent. Comparing our obtained data per task (see Table 2), our results show that the number of respondents able to accurately select an icon and drop it into every single blank state obtained the lower limit of 0.3795 and the upper limit of 0.6622.

**Table 2.** Summary of test results on icon recognition.

Test Type	Test Result
Individual Icon Recognition	The 95% confidence interval for the proportion of potential participants who could predict icons' meaning and correctly place icons in the requirements life cycle stage was .3795 to .6622.
Multiple Icon Recognition	The 95% confidence interval for the proportion of potential participants who could correctly interpret multi-icons' meaning was .7331 to .9306.
Compound Icon Recognition	The 95% confidence interval for the proportion of potential participants who could correctly construct iconic sentence from a given icon was .7543 to .9540.

We observed that multiple icons were recognized fairly well (lower level of 0.7331 and upper level of 0.9306). In terms of recognition of the 14 icons, roughly 17 percent of respondents had misunderstandings. Compound icon recognition achieved a correct prediction accuracy of 0.7543 at the lower limit and 0.9540 at the upper limit.

#### 4.2.2. Icon Satisfaction/Usability

For the aspect of usability, we collected information about satisfaction, effort expectations and attitudes toward using the system [41]. We included scales to describe users' perceived competence in mastering icon-based language tasks. We complemented the dimension of usability with satisfaction opinions [42] (with variables such as complicated/easy or unpredictable/predictable). The experiment ended with three questions on the participants' overall preferences, which directly point to icon-based language usage. The first question, related to the satisfaction of icon-based language, was measured on a 4-point scale (0: no opinion, 1: dissatisfied, 2: satisfied, and 3: very satisfied). The second question regarded attitudes toward using the system and was assessed on a 3-point scale (0: no opinion, 1: will not use icon-based language, and 2: will use icon-based language). The third question, which focused on attractiveness, was appraised by a 5-point scale (0: no opinion, 1: make communication more clear, 2: the structure was presented clearly, 3: intuitively understood, and 4: easy to use). The results in Table 3 describe three primary aspects: (1) satisfaction level, (2) likelihood for further use of icon-based language, and (3) effort expectations.

Satisfaction was based on the degree to which participants perceived that icon-based language is beneficial for developers and other stakeholders. The overwhelming percentage of respondents was "satisfied" (69.57%).

However, icon-based language has significant differences in attitudes when it comes to using the system. According to the figure shown in Table 3, more participants plan to use icon-based language (39.13%) than not (17.39%). However, those with a positive attitude were still less than those with no opinion (43.48%).

The question on effort expectation gauges the degree of ease associated with the use of a system. These results explained subjective perceptions of usability (e.g. "I find icon-based language to be intuitive" or "I find icon-based language easy to use", etc.). The intuitive aspect (32.61%) was perceived to be the highest. According to respondents, making communication clear (28.26%) was the second notable feature supported by icon-based language.

**Table 3.** Measurement of subjective scales of users' satisfaction (46 replies out of 48).

Measurement Type	Rating	
	Scale	Portion
Satisfaction	0 = no opinion	13.04
	1 = dissatisfied	17.39
	2 = satisfied	69.57
	3 = very satisfied	0.00
Attitude toward using the system	0 = no opinion	43.48
	1 = will not use IL	17.39
	2 = will use IL	39.13
Effort expectation/Attractiveness	0 = no opinion	15.22
	1 = make communication more clearly	28.26
	2 = the structure presented clearly	13.04
	3 = intuitive understanding	32.61
	4 = easy to use	10.87

## 5. Discussion

The experiments presented in this paper reveal that user can indeed sufficient recognize icons very well when a group of related icons is presented. However, when individual icon is portrayed separately it seems practitioners getting misunderstood.

Here we summarized our experiments by excluding individual icon interpretation. Both of our experiments show promising results, which generally support our approach. In terms of Hypothesis 1, icon-based language proved to be significantly well recognized by participants across all tasks. Specifically, compound icon recognition (which imitated the goal-oriented model) was correctly recognized by a majority of respondents (85%).

Here we reported our experiments by including individual icon interpretation. While derived results are encouraging, they also indicate a need for improvement for individual icon recognition; in practice, training on RE components (such as process, activities and life cycle to users) could advance users' knowledge and skill. Therefore, the findings confirm a prerequisite need for users to capture RE knowledge.

Hypothesis 2 was substantiated by the results of our questionnaire, where participants described icon-based language usability: icon-based language is successful when it comes to making users satisfied, enriching communication, and supporting intuitive understanding. Regrettably, icon-based language fails in terms of perceived ease of use: only 10.58 percent of respondents felt that it is easy to use. This failure to offer stakeholders an easy-to-use approach could have several reasons. First, the survey did not cover icon-based language as a whole, but presented only a part of icons' attributes. As a consequence, respondents might not have been able to comprehend how icon-based language can be used and how it benefits RE stakeholders. Secondly, it was not our intention in this study to emphasize icon designs. Therefore, a limitation was that all visual vocabularies in this paper were gathered from existing ones and only used to represent concepts and ideas. At this stage, we were not concerned with their appearance in terms of what they represented. This task must be taken up by designers, who are experts in that area. Design relies enormously on cultural experience and cognitive effectiveness.

It is worth noting again that the adaptive preference framework (Section 2.3) does not yet implement in this phase because we attempt to make our concept available for only Finnish participants. Only one variable, nationality, was taken into account for pattern presentation, therefore, all practitioners view icon visualization in the same fashion since we assume that all practitioners are Finnish. Nevertheless such framework is needed to precisely define all necessary variables that could be supportive information for acquiring prospective participants as well as for further development.

## 6. Conclusion

It has long been acknowledged by industry experience and research that RE is a crucial factor that contributes to software success or failure and that icons are frequently used to supplement texts and overcome language barriers. While more and more RE techniques offer visual notations, they often use abstract shapes with clearly conventional meanings, which must be learnt. Additionally, little effort has been made to correlate icons with such techniques. This study set out to determine whether icons are capable of representing RE concepts. One of the more significant findings to

emerge from this study is that icons that characterize requirements attributes such as priority, status and stakeholders can be correctly recognized by users. This study also finds that respondents conceive icons as an intuitive medium for communication.

As with most novel approaches, our study of icon-based language has opened up possibilities for new and exciting future research. It is recommended that further research be undertaken in the following areas: (1) developing user preference framework for other nationalities, (2) implementing demonstrators for icon-based that can be used by multi stakeholders, and (3) assessing implemented tools in terms of their benefit and ease of use for a range of stakeholders.

## References

- [1] Pohl, K., *Requirements engineering fundamentals principles and techniques*, Springer Berlin, 2010.
- [2] Wiegers, K. E., *Software Requirements*, Second Edition, Microsoft Press USA, 2003.
- [3] Thanasankit, T., & Corbitt, B., Understanding Thai culture and its impact on requirements engineering process management during information system development, *Asian Academy of Management Journal* **7(1)** (2002), 103-126.
- [4] Carrillo de Gea, J. M., Nicolas, J., Aleman, J. L. F., Toval, A., Ebert, C., & Vizcaino, A., Requirements engineering tools, *Software, IEEE* **28(4)** (2011), 86-91.
- [5] Mathiassen, L., Tuunanen, T., Saarinen, T., & Rossi, M., A contingency model for requirements development, *Journal of the Association for Information Systems* **8(11)** (2007), 569-597.
- [6] Al-Rawas, A., & Easterbrook, S., Communication problems in requirements engineering: A field study. Proceedings of the First Westminster Conference on Professional Awareness in Software Engineering, Royal Society, 1996.
- [7] Coughlan, J., & Macredie, R. D., Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering* **7(2)** (2002), 47-60.
- [8] Sutton, D. C., Linguistic problems with requirements and knowledge elicitation. *Requirements Engineering* **5(2)** (2000), 114-124.
- [9] Lang, M., & Duggan, J., A tool to support collaborative software requirements management. *Requirements Engineering* **6(3)** (2001), 161-172.
- [10] Kimball, R., Is ER Modeling Hazardous to DSS?. *DBMS Magazine*, 1995.
- [11] Moody, D. L., Heymans, P., & Matulevicius, R., Improving the effectiveness of visual representations in requirements engineering: An evaluation of i\* visual syntax, *17th IEEE International on Requirements Engineering Conference*, pp. 171-180, 2009.
- [12] Bendraou, R., Jezequel, J., Gervais, M-P., & Blanc, X., A comparison of six UML-based languages for software process modeling, *IEEE Transactions on Software Engineering* **36(5)** (2010), 662-675.
- [13] Moody, D. L., & Hillegersberg, J. V., Evaluating the visual syntax of UML: An analysis of the cognitive effectiveness of the UML Family of diagrams, *Software language engineering*, pp. 16-34, 2009.
- [14] Morris, S., & Spanoudakis, G., UML: An evaluation of the visual syntax of the language, *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pp. 1-10, 2001.
- [15] Barber, W., & Badre, A., Culturability: The merging of culture and usability, *Proceedings of the Conference on Human Factors and the Web*, 1998.
- [16] Callahan, E., Cultural similarities and differences in the design of university web sites, *Journal of Computer-Mediated Communication* **11** (2006), 239-273.
- [17] Corbitt, B., & Thanasankit, T., A model for culturally informed web interfaces, In J. D. Haynes (Ed.), *Internet management issues* IGI Publishing, 2001.
- [18] Reinecke, K., & Bernstein, A., Improving performance, perceived usability, and aesthetics with culturally adaptive user interfaces, *ACM Transactions on Computer-Human Interaction* **18(2)** (2011), 8-29.
- [19] Reinecke, K., & Bernstein, A., Knowing what a user likes: A design science approach to interfaces that automatically adapt to culture. *MIS Quarterly* **37(2)** (2013), 427-453.
- [20] Burgmann, I., Kitchen, P. J., & Williams, R., Does culture matter on the web? *Marketing Intelligence & Planning* **24(1)** (2006), 62-76.

- [21] Heimgärtner, R. H., Andreas., Towards cross-cultural adaptive driver navigation systems, *Usability Symposium, Austrian Computer Society* **198** (2005), 53-68.
- [22] O'Neill-Brown, P., Setting the stage for the culturally adaptive agent, *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents*, 1997.
- [23] Hevner, A. R., March, S. T., Park, J., & Ram, S., Design science in information systems research, *MIS Quarterly* **28(1)** (2004), 74-105.
- [24] Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S., A design science research methodology for information systems research, *Journal of Management Information Systems* **24(3)** (2007), 45-78.
- [25] Cerpa, N., & Verner, J. M., Why did your project fail? *Communication of the ACM* **52(12)** (2009), 130-134.
- [26] Khanom, S, Icon-based language in the context of requirements engineering. *REFSQ Requirements Engineering: Foundation for Software Quality*, pp. 215-222, 2013.
- [27] Khanom, S., Heimbürger, A., & Kärkkäinen, T., Icon-based language in requirements development. *22nd European Japanese Conference on Information Modeling and Knowledge Bases*, pp. 20-25, 2012.
- [28] Khanom, S., Heimbürger, A., & Kärkkäinen, T., Icon-based language: Auxiliary communication for requirements engineering. *International Journal of Engineering Science and Technology* **5** (2013), 1076-1082.
- [29] Khanom, S., Heimbürger, A., & Kärkkäinen, T., Icon-based language in the context of requirements elicitation process, *23rd European Japanese Conference on Information Modeling and Knowledge Bases*, 2013.
- [30] ISO/IEC 9126: IT- Software Product Evaluation – Quality characteristics,” International Organization for Standardization, 1991.
- [31] Heimbürger, A., & Kiyoki, Y., Pictorial symbols in context - A means for visual communication in cross-cultural environments. *Proceedings of the IADIS International Conferences, IADIS Press*, pp. 463-467, 2010.
- [32] Heimbürger et al, Communication across cultures in the context of multicultural. Software Development, Reports of the Department MIT. Series C. Software and Computational Engineering, 2011.
- [33] Heimbürger et al., Intelligent icons for cross-cultural knowledge searching, *Information Modelling and Knowledge Bases XXIII* **237** (2012), ISO Press Amsterdam, 77-89.
- [34] Aykin, N., Usability and Internationalization of Information Technology, Lawrence Erlbaum Association, Inc., Publisher, 2005.
- [35] Hofstede, G., Cultures and organizations: Software of the mind, New York: McGraw-Hill, 1997.
- [36] Ackerman, S.K., Mapping User Interface Design to Culture Dimensions, *Proceedings of IWIPS*, Austin, Texas, pp. 89-100, 2002.
- [37] Shen, et al., Towards Culture-Centred Design, *Elsevier-Interacting with Computer* **18** (2006), pp. 820-852.
- [38] Marcus A. and Gould E.W, Cultural Dimensions and Global Web UI Design, *Interactions* **7-4** (2000), pp. 32-46.
- [39] Majchrzak, A., Wagner, C., & Yates, D., The impact of shaping on knowledge reuse for organizational improvement with wikis, *MIS Quarterly* **37(2-Appendices)** (2013), 1-12.
- [40] Khanom, S., Heimbürger, A., & Kärkkäinen, T., Icons: Visual representation to enrich requirements engineering work, *Journal of Software Engineering and Applications* **6(11)** (2013), pp. 610-622.
- [41] Venkatesh, V., Morris, G. M., Davis, B. G., & Davis, D. F., User acceptance of information technology: Toward a unified view, *MIS Quarterly* **27(3)** (2003), 425-478.
- [42] Hassenzahl, M., Platz, A., Burmester, M., & Lehner, K., Hedonic and ergonomic quality aspects determine a software's appeal, *Proceedings of the International Conference on Human Factors in Computing Systems*, pp. 201-208, 2000.

# Knowledge Support for Software Processes

Michael Alexander KOSINAR<sup>1</sup>, Jakub STOLFA<sup>2</sup>, Svatopluk STOLFA<sup>3</sup>

<sup>1,2,3</sup>*Department of Computer Science*

*VŠB – Technical University of Ostrava, Faculty of Electrical Engineering and  
Computer Science*

*708 33, Ostrava – Poruba, Czech Republic*

**Abstract:** In the last decades we have got used to software applications and services being everywhere and working for us, improving our lives. Even though sometimes they fail to work as desired. The situation we experience nowadays is often characterized as the second software crisis and it is caused by many root causes. One main root problem is an old one – it is an insufficient specification of requirements and processes to be executed in software development. Even though the computer science world offers many specification methods, standards, generic software processes, best practices and languages, the problem is still here. The research presented in this work proposes utilization of formal methods and knowledge bases as a key solution for software requirements processing and quality control; the goal of the research is to describe particular sub-processes of software development process and their optimization with formalisms like Web Ontology Language (OWL). We will propose and develop complex software process modeling methodology that would combine semi-formal and formal approaches with forward and reverse process engineering (process mining) that would be used easily as well-known semi-formal approaches.

**Keywords:** software process, requirements management, quality assurance, formal methods, UML, OWL, enterprise framework, process mining

## Introduction

Software development market has been facing the challenges of rapidly changing environments that change quickly from closed and centralized approaches to open and distributed with background business processes being more complex. Because of that, organizations and companies are paying greater attention to business process management support so it is able to adapt new complex business needs and environments.

Traditional approach to process modeling (or enterprise modeling) is to model and manage the process based on a structure, behavior and functional description of a company. Process modeling approaches based on traditional specifications work well for stable or simple business processes; however it is not sufficient for business processes that could suffer from rapidly changing environment and may ask for more dynamic approach that wouldn't lack adaptability [1, 11, 14].

With rising need of adequate adaptability and flexibility in business processes many research groups have been investigating knowledge-based methodologies, adaptive process techniques, etc. [12, 26]. Some of these approaches utilize explicit representation of alternatives in the process or process redesign to deal with exceptional situations and changing conditions; nevertheless they offer only limited adaptability

and may cost a fortune which make them quite impractical for real use. Another solutions involve utilization of formal systems, knowledge bases and mathematical models (e.g., Discrete-Event Simulation, System Dynamics, etc.). Implementation of formal rules and facts, the process model and its execution may be more adaptive to unexpected situation and events.

For example in a process of requirements management, all requirements should be processed in iterations as defined in software process best practices [37]; yet, as said before, systems are growing more complex and their environment may be unstable and change in time. Such changes don't have to necessarily change the requirements management process as whole but may affect internal functions and rules and format of inputs and outputs of the process. Without swift reaction, the process instance could generate invalid requirements in wrong format leading to bad results.

Conventional semi-formal approaches and forward process engineering may lead to misunderstandings and errors due to their lack of formal power and formal knowledge-based systems are not easy to use in practice because of their complexity. A creation of a simple model may become a challenge with traditional modeling techniques in unstable environment. Our research focuses on software process modeling novelization that will overcome the limitations of existing approaches.

This paper demonstrates the implementation of these mechanisms. Combination of Unified Modeling Language, Web Ontology Language, enterprise scheme (e.g. Zachman Framework) and methods based on both forward and reverse engineering are used to implement them. The intention was to develop a methodology that will use knowledge representation language to model a reality. Semi-formal models (new or existing) will be extended with formal models of the reality. Finally the models are about to be embedded to the enterprise scheme of the organization. The goal of the research is to solve or prevent possible loss of the particular information from the reality, lack of understanding, unstable environment and changing markets; mostly the issues occurring when we model directly in any semiformal or formal language using traditional techniques.

The thesis is organized as follows:

- Section 1 covers the state of the art in the field of the research
- Section 2 introduces the basic theories and proposal of the methodology with a simple software process example hierarchy and analysis
- Section 3 discusses the utilization of reverse engineering and process mining methods and the support of the formal SP methodology
- Sections 4 and 5 conclude the benefits of the approach and a summary of the work and outlines future work.

## **1. Knowledge support to software process modeling – State of the art**

Business processes represent the core of company behaviour. They define activities that companies (i.e. their employees and structures) perform to satisfy needs of their customers. For a definition of the term business process, we use the definition of Workflow Management Coalition [40]: “*Business process* is a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the structure defining functional roles and relationships.” A



process of information systems or software products development is called the software process. The software process is also a kind of business process but it has its specific aspects [20].

There exist many modeling techniques for process modeling as is mentioned in Vergidis [36]. On the other hand, the software process is quite specific [28] and it has been characterized as “the most complex endeavor humankind has ever attempted” [6]. However, software process could be modeled formally; the main objectives and goals for software process modeling are defined in Curtis [7]:

- *facilitate human understanding and communication*
- *support process improvement*
- *support process management*
- *automated guidance in performing process*
- *automated execution support*
- *simulation*

Software engineering is a discipline that is involved in software process development and maintenance. A process development could be divided into activities of different kinds as is defined by the process life-cycle engineering. Description of software process life cycle activities is provided in [10].

The research initiative is the modeling of software processes. There are discussed benefits of knowledge based approach in [31] and another approaches are compared in [34]. Primary intention is to develop simple and stable modeling foundations that will enable:

- *integration* of different approaches – avoiding duplicities when modeling by one approach and then switching to another; i.e. model extensions that would offer practical extension of one well-known modeling approach with formal knowledge-based approach
- *transformation* between different types of model approaches;
- *iterative* creation of models - the model can be modeled from the abstract viewpoint and then refined;
- *automatic composition* of workflow base on desired workflow;
- *different views or useful information* for modelers during model creation by defined queries – e.g. probable resource allocation deducible from model dependencies;
- *model refactoring* – structural changes in models is painful process and it should exist similar refactoring possibility as is in a programming.

### 1.1. Process modeling and UML

To propose a proper formal modeling discipline, we must follow the concepts of Workflow Management Coalition [40]. The basic concepts defined by WMC are *Workflow*, *Process* and *Activity*.

*Workflow* is the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.

*Business Process* is a set of one or more linked procedures or activities, which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.

The *process definition* consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc. The process definition may contain references to sub-processes, separately defined, which make up part of the overall process definition.

A lot of process models associate the concept of process with the concept of activity. For instance *Activity diagrams* are pattern on the concept of activity. They are the specialization of Unified Modeling Language (UML) state-machine diagrams and are based on the Petri net semantics [5, 29]. Activity diagrams describe graphical representations of processes of stepwise activities and actions with support for choice, iteration and concurrency. In UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagrams elements are for instance: *Activity*, *Structured activity*, *Action*, *Object*, *Datastore*, *Decision* etc.

### 1.2. Visual Process Modeling Language and Business Process Model Approaches

Visual Process Modeling Language (VPML) is another graphic language that supports a special process definition. The process model built in VPML can be simulated. It is theoretically proved that VPML is equivalent Petri-Net [29].

Business Process Model and Notation (BPMN) is the graphical tool for the process specification. It is based on the similar principles as activity diagrams [3].

However the first named is not that spread across the organization and the second is not designed to diagram the detailed steps (tasks) in a procedure; thus we started to use a combination of well-accepted UML and OWL.

### 1.3. Web Ontology Language

There are several solutions of *knowledge base* creation. These are based on formal languages like *Cycl* [22], *Casl* [25], *Object Constraint Language (OCL)* [37] and logical languages like *Prolog* [19, 22]. When we choose a representation for the processes, we must consider its attributes and balance between expressive power, rigor and ease of use of a representation [32].

An ontology is a data model that represents a set of concepts within a domain and the relationships among those concepts. It is used by machines to reason about the objects within that domain. RDF (Resource Description Framework) is an XML based syntax standard where one can define statements about a resource in the form of subject-predicate-object expressions called triples [4]. RDF Schema (RDFS) defines the semantics of any particular domain with which concepts can be readily described and referred by RDF. Both RDF and RDFS have limited expressive power.

*Web Ontology Language (OWL)* provides a more expressive ontological description of complex relations between concept pairs than RDFS does. RDFS elements can be used to define a concept in terms of a class and assigned properties in OWL [38].

A knowledge base may use ontology to specify its structure (entity types and relationships) and its classification scheme. Ontology, together with a set of instances of its classes constitutes a knowledge base that includes:

- a set of concepts, properties, and the relationships among them in a specific application domain,
- high-level abstraction of rules in the form of constraints,,
- semantic service descriptions and
- a semantic model of event context.

An ontological knowledge base builds the metadata needed by a procedure to understand its environment and status, reason about them; It can be used also to schedule tasks with optimized resources while complying with business policies and compliances.

#### 1.4. *Software process reverse engineering approach*

Business process execution could be, and usually is, supported by software system [8]. It is because of that the software systems that are used to support daily business are in last decades more a more process oriented. This area went through the revolution where systems were switched from data oriented solutions to process oriented solutions. Process oriented system is the only way how to control the performed process and its activities. This switch form the data oriented systems to the process oriented systems bought companies power to control and check the enactment of the processes and the resources that are involved [35].

On the other hand, there is a Business process management (BPM). BPM [39] can be defined as whole business company process management and optimization. Its concern is on the process improvement and its alignment to the needs of clients. BPM lifecycle consist of design, modeling, execution, monitoring and optimization. It means that the BPM take care of the composition, enactment and analysis of the operational business processes.

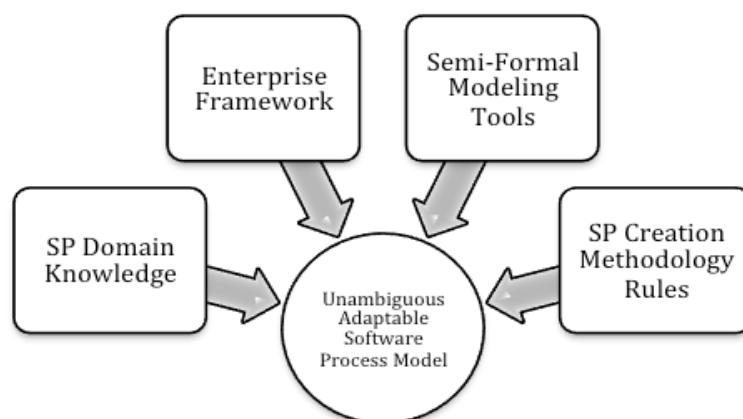
Business process definitions are sometimes quite complex and allow many variations. All of these variations are then implemented to supportive systems. If you want to follow some business process in a system, you have many decisions and process is sometimes lost in variations. Modeling and simulations can help you to adjust the process, find weaknesses and bottleneck during the design phase of the process. Sometimes you guess or know the patterns and occurrence probabilities of variations that are used during the execution phase. However, not even modeling and simulation of the processes can tell you, how processes are really enacted in the system, what is e.g. the perceptual usage of the variations and whether some variations are enacted at all. If you want to analyze the real usage of the system, recognize its weaknesses, bottlenecks or strength of the real data, you have to know how the process was followed and executed in reality. Process mining is an approach that is used for the analysis of real enactment of the processes. Process mining uses logs of real process enactments to analyze the process itself. Process mining can answer you the question, how the process was really executed, which variations were used and what are the probabilities of the enactment of each process variation [1].

Such an approach gives us the possibilities to discover the process from the bottom. It is possible to use data created by the process execution for discover,

modeling and adjusting purposes. On the other hand, process modeling gives us view and process discovery from the top. It means that the domain of the process is discovered first and then is process modeled and described in the knowledge base.

Software process reverse engineering, respectively part of the process mining, is the supplement for the knowledge support in the process modeling in the meaning of process discovery from existing systems and its comparison. The comparison of the modeled process and the real process execution helps to perform continuous process improvement.

The research is supported by cooperation with local software companies and the methodology's best practices and procedures are based upon the real needs of healthcare information system development. The goal, as outlined at the beginning, is to improve the software process modeling in companies. Although we've presented above that many approaches, tools and techniques exist, we propose to base the methodology foundations on commonly used tools well-known to software production enterprise; *Unified Modeling Language* used for process modeling, *Zachman Framework* [30] for organizing the process artifacts and sub-processes and *Web Ontology Language (OWL)* for depicting the formal description of the process. Such approach will improve the conditions of acceptance on one hand and many existing editing tools on the other.



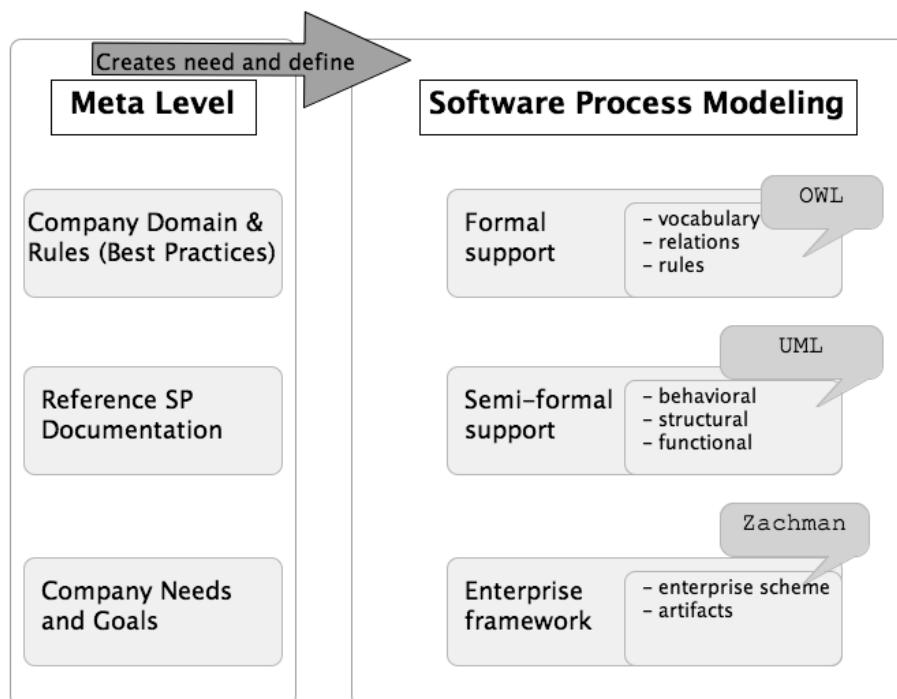
*“Our hypothesis is based on experiments and observations of real software processes; If we can support the software process modeling with formal methods like OWL knowledge bases, semi-formal methods like UML activity diagrams and process mining methods, then we could get unambiguous process models, adaptable to changing environments and markets but yet preserving ease of use and readability of models created in UML and embedded into enterprise scheme like Zachman framework”*

## 2. Software Process Knowledge Framework – Modeling discipline

Software process business rules are used to represent business policies and logics in a context of software development companies. Behavioral, functional and structural

business models are often delivered in *UML diagrams* [5]. *Ontologies* provide a formal semantic knowledge representation model to capture business objects as concepts and business rules behind the described business objects. Properties, constraints, and relations are defined to describe high-level business rules. In the proposal, we define an ontological knowledge framework of a software process; i.e. extension of the software process modeling with knowledge base [16]. To provide a detailed view of all the entities involved in software requirements management, this approach partially covers the knowledge domain of enterprise software development.

Therefore, it allows this Software Process Knowledge Framework to capture all necessary knowledge for a more complex scenario involving software development, legislation policies and others – see figure 1.



**Figure 1.** Software Process Knowledge Framework

### 2.1. Example of software process

The proposal of Software Process Knowledge Framework and Methodology is tested on generally used example and well known process of requirements management. There are defined main workflow activities (see figure 2): Analyze the Problem, Understand Stakeholder Needs, Define the System, Manage the Scope of the System, Refine the System Definition and Manage Changing Requirements. Every workflow activity is internally described in greater detail and consists of tasks; detail of tasks may vary in different companies or even departments. We also know that internal rules serve as control mechanism of the activity, each task has defined input and output artifacts and is performed by specific responsible roles.

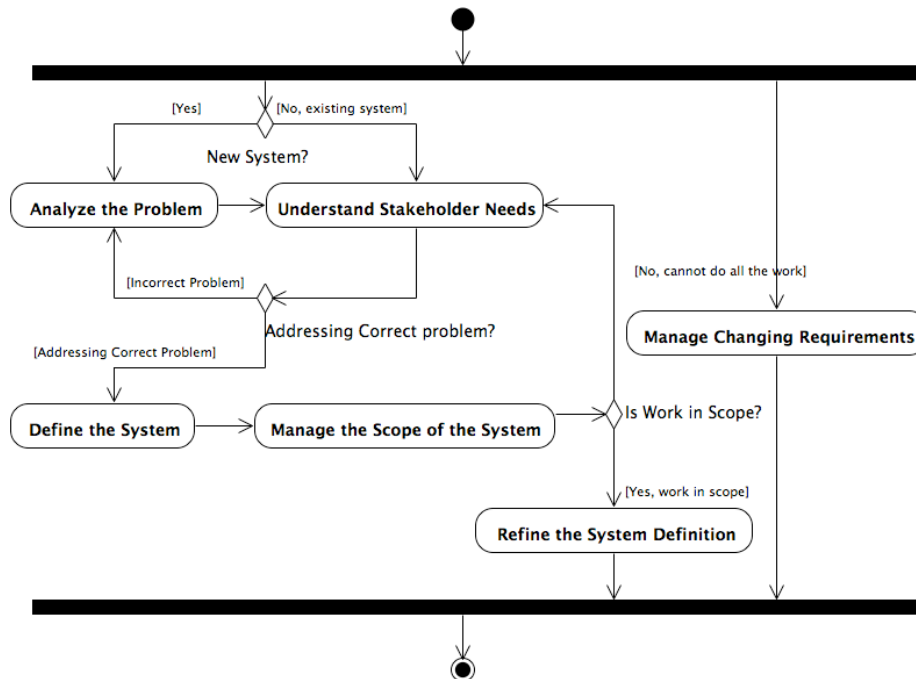


Figure 2. Activity diagram of requirements management workflow (RUP)

## 2.2. Modeling constructions

As a part of modeling technique, this framework includes explicit declaration of groups of elements, description of group relations, some specific conditions, definition of individuals etc. The granularity is based on process levels illustrated in table 1. Each lower level can be separated in greater detail up to atomic rules of certain tasks. This granularity model provides a base of software processes semantics. A core part of the methodology is a central semantic repository with domain knowledge based on reference software processes (e.g. Rational Unified Process). This approach allows domain specialists to compose or alter the process knowledge base and deploy it easily in a flexible way.

The general process of software process modeling with knowledge support consists of following core workflows (see figure 3):

- (1) Process observation in organization
- (2) Software process modeling
- (3) Knowledge Modeling
- (4) Compilation of Software Process and Knowledge Models
- (5) Model implementation in the organization
- (6) Optional – Refine the parameters of observation and re-initialize it

The proposed technique gives us powerful knowledge support to the (2) Software process modeling workflow even though the procedures behind are well known and always should be based on best practices. In order to support the workflow properly, the knowledge base must be defined in a clear valid way. The base is composed of

terms, rules and patterns common to software process but only well-formed knowledge can be used to support the modeling. We use three fundamental functions of abstraction for the system modeling; these are defined in Smith [33] and Machado [23].

- a) *Aggregation* – entity containing other parts from modeled domain is represented by one entity in the model;
- b) *Classification* – class of similar entities and their features is identified;
- c) *Generalization* – different set class of entities are unified into one class with similar properties.

<b>Level 1</b>	Enterprise Process Level	Software Process in the Company
<b>Level 2</b>	Core Process Level (within enterprise process)	Requirements Management; Quality Management; ...
<b>Level 3</b>	Sub-Processes Level (within core process)	Analyze the Problem; Define the System; ...
<b>Level 4</b>	Tasks Level (within subprocesses)	Develop Vision; Capture a Common Vocabulary; Find actors and Use Cases
<b>Level 5</b>	Internal Rules Level (within tasks)	Identify user groups; identify external systems; create use case diagram

**Table 1.** Software Process Levels Reference House

Based on the abstraction functions, the framework supports following functions – classification is mainly represented by the relation member-class; generalization is a subclass-class relation (IS-A relation); aggregation is directly supported but it is expressed by the special kind of association:

- definition of separated software process specifications (models, ontologies);
- definition of mentioned classes (sets);
- definition of associations among classes (relations);
- constraints definition;
- definition of macros or user functions

#### *Definition of separated specifications*

Specification represents defined model by concrete elements, constraints and relations between them. One input could be a basic domain knowledge defined in OWL, the other one could be semi-formal model described in UML diagrams. Discussion on the importance of separate specifications existence is in [17]. Possibility of importing any other specification is also involved. It enables us to define separate meta-model and use it for the definition of many independent models afterwards. Every rule in imported specification should be valid in importing theory [18]. On the other hand, rules in the independent specification can be in conflict that can lead to invalid specifications. This construction is presented in OWL. Example in OWL looks like:

```

<owl:Class rdf:ID="RequirementsSpecifier">
  <rdfs:subClassOf rdf:resource="#Role"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasTasks"/>
      <owl:someValuesFrom rdf:resource="#SoftwareRequirementsSpecification"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

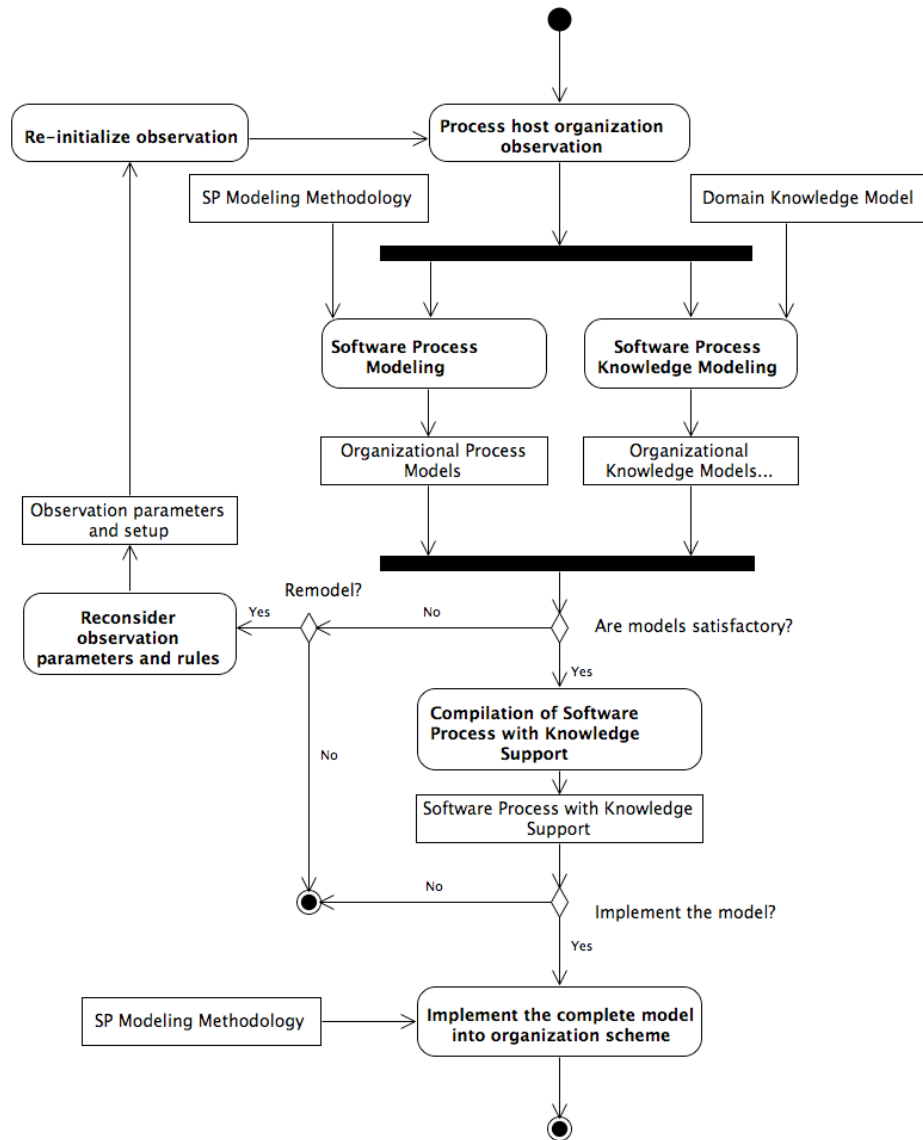


Figure 3. Software Process Knowledge Framework application

*Definition of classes (sets)*

Definition of classes (sets – elements defined in meta-model and used during modeling), [9]:

$$\textit{Artifact} \subset \textit{Thing}$$

Set “Thing” contains every item in modeled universe. Construction of subset is used there for definition of specific classes.



Classes can be used for the definition of particular objects (model from ISPW-6 [13]) with definition of membership:

$$\text{UseCase} \in \text{Artifact}$$

Specific “Use Cases” created for different instances of process can be defined in this way:

$$\text{useCase\_gatherRequirements}_1 \in \text{UseCase}$$

Multilevel classification is used here so that item that is member of a class is also class and may contain another member classes. Such definitions enable us to work with classes as elements. This type of thinking allows us to use multiple classifications. For example we need to define a condition that specific “Artifact” is connected by the relation “is modified by” exactly with one “Role” – on the other hand more people in this role could be responsible for the specific “Processes”.

#### *Definition of associations (relations)*

The static view of the process always contains some classes, but separate classes have to be connected somehow. Otherwise identified classes would be nothing else than the definition of terms. We need to define relations to describe a connection of one class to another. Classes and their relations together form a basis of a static description of the modeled system. For example the definition of the simple relation – modification of artifact by specific role:

$$\text{isModifiedBy} \subset \text{Artifact} \times \text{Role}$$

Another necessary construction is an ability to define attribute that is nothing else then relation. Definition that every “Artifact” has a “Textual Description” could be expressed:

$$\text{Description} \subset \text{Artifact} \times \text{String}$$

Assignment of description to specific “Artifact” is nothing else then definition that concrete relation exists as a member:

$$\langle \text{"This artifact defines a set of use case instances"} \rangle \in \text{Description}$$

#### *Constrain definition*

Some specific relations and classes have to satisfy some constraints. Example of a constraint could be the definition that every “Artifact” should be in the relation “Artifact\_output” with at least one “Department”:

$$\text{Artifact}_{\text{output}} \subset \text{Artifact} \times \text{Department}$$

$$\forall x \left( x \in \text{Artifact} \Rightarrow \left( \exists r (r \in \text{Department} \wedge \langle x, r \rangle \in \text{Artifact}_{\text{output}}) \right) \right)$$

Existential and general quantifiers and logical operators are used for such kind of constraints.

#### *Macro or user function definition*

More complicated expressions are sometimes needed and used to express relatively simple combinations of basic constructions. The constraint defined previously is an example. It can be required for another relations and it may be impractical to define it in such way. Solution can be to use macros or user functions definitions to simplify the expressions and enable comfortable construction and usage of complicated expressions:

$$\text{someItemOnLeft}(\text{relation}) \stackrel{\text{def}}{=} \text{relation} \in \text{Domain} \times \text{Range} \\ \Rightarrow \forall d \left( d \in \text{Domain} \Rightarrow (\exists r (r \in \text{Range} \wedge \langle d, r \rangle \in \text{relation})) \right)$$

Then only this fact is defined during the modeling:

$$\text{someItemOnLeft}(\text{Artifact}_{\text{output}})$$

### 2.3. Requirements Management Model

Following part covers the example of the general modelling process. The example is based on partial software process based on the reference framework of Rational Unified process – Requirements Management workflow. Combination of proposed tools, Unified Modeling Language, Web Ontology Language and enterprise scheme provide enough expression power for capturing all classes, data properties and object properties of our determined ontology including the embedding into the enterprise.

#### *Initial phase – modeling inception phase – domain and process observation*

The initial phase, as defined in the proposal, covers observation of the company, real software business needs including possible utilization of reference software process. This part takes an input of company process requirements, responsible roles, software products domain and consults them with reference software process.

#### *Elaboration phase – model creation phase*

The second described phase is more complex to perform as it is proposed as two separated parallel activities, yet related to each other. Because the process software process modeling in the forward engineering approach should be modeled out of building blocks extracted from knowledge base, the domain modeling should precede at least few steps the very process modeling. Thus the first steps of this phase will cover the domain knowledge modeling.

#### *First step - entities*

First step of the work would be a creation or re-usage of entities. In our case some of the entities in the hierarchy are:

- Class Process – Requirements management*
- Class Activity – Analyse the Problem, Refine the System Definition, Manage Changing Requirements, ...*
- Class Role – Technical Staff, Support Staff*
- Class Technical Staff (roles) – Requirements Specifies, System Analyst, ...*
- Class Support Staff (roles) – Consultant*
- Class Resource – Employee, Device, ...*
- Class Task – Find Actors, Find Use Case*

### *Second step – data properties*

To complete the particular entities description, we must define data properties. Data properties shouldn't be defined only to complete the entities description in a proper way, but could be used also to define KPIs metrics related properties. For example Office can be described with properties capacity (amount of people interpreted as integer or set of unidentified persons), actual used capacity, lease price (local currency); Employee can be described with wage, years of experience, etc. The properties are set with values after creation of the class instances.

### *Third step – object properties*

Third step of the process is to describe the object properties of all entities. The procedure is quite similar to data properties description but in this case we have to focus on relations with other entities. First step would be to identify all relevant object properties then we have to make connection between entities with them.

*isModifiedBy* – relation between an artifact and a role; e.g. use case is modified by requirements specifier;

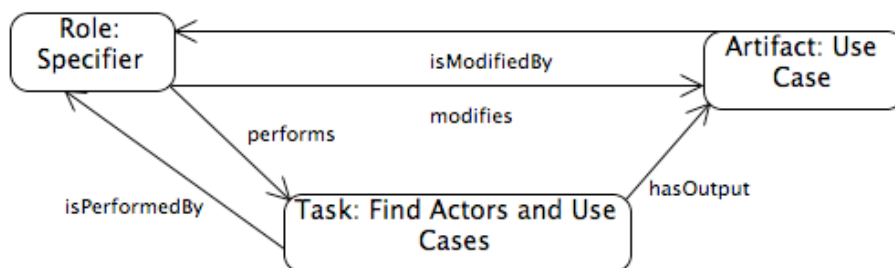
*isPerformedBy* – relation between process, sub-process or an activity and a role;

*performs* – relation between role and process;

*consistsOf* – property of process entities creating relations with activities and activities with tasks; e.g. Requirements Management consists of Define the System and Define the System consists of Find Actors and Use Cases;

*isParallelWith* – is a relation between processes, sub-processes, activities and tasks; gives us a definition that entities in the relation are executed in parallel way;

For example, the element Use Case has more complex definition. We cannot determine everything using only data properties, thus we will use object properties. Use Case artefact, Find Actors and Use Cases task and Requirements Specifier role entities have to be described in advance with their particular data properties. Then we have to model relation between the entities with object properties *isModifiedBy*, *isPerformedBy* and *performs*. The following example on figure 4 demonstrates a definition of that element and its data and object properties:



**Figure 4.** Modelled properties in graphical representation

*Compilation and implementation phase – model adjustment and implementation into enterprise scheme*

The knowledge base is described in XML with OWL language and the process models are in UML diagrams, created from building blocks acquired from the knowledge base. To complete the modeling process, we have to fit and implement the model into the business by using certain enterprise framework. In our case we decided to use IBM's Zachman Framework. The procedure of the scheme creation is based on filling the matrix with created models to answer critical questions *why, how, what, who, where* and *when* in specific scopes like contextual, conceptual, logical, physical and detailed. With valuation of matrix values with vocabulary in OWL knowledge base, diagrams in UML and rules specification and details described with OWL we get highly organized and readable enterprise model of software process tailored to organization's needs.

### **3. Reverse engineering approach in correlation with software process knowledge framework**

Reverse engineering approach gives us opportunity to review and discover the process model from the data of the real process. In correlation with software process knowledge framework is possible to support knowledge framework in the two ways:

1. Check the correctness of the process model by comparison of the modeled process and the real process execution. It means that if the process model was set up by software process knowledge framework, it is possible to check if the process is followed correctly or if there are some deviations. This comparison helps in the continuous process improvement. There you can see and compare if the modeled process is really followed in reality and how.
2. At the beginning of the execution by software process knowledge framework it is possible to discover the current process if there is one. If there is already some undefined process supported by some software system, reverse engineering can help us to understand the domain and actual state of the software process.

#### *3.1. Check the correctness of the process model*

We need to follow several particular actions to execute comparison of the modeled process and the real execution of the process. Figure 7 depicts process of the checking correctness of the process model. Particular steps are described below.

##### *Extraction of IS data logs*

This is important step of reverse engineering. If it is not possible to obtain solid data from the logs, we cannot continue in the reverse engineering.

We have to collect all possible logs from the systems that are related to the obtained process. We need to ensure that we take a lot of data to cover whole process. That means if one case of the process execution should take at least a year, we have to take the logs for whole year. Usually it is more than one data file, so that we have to link the files together.

If we are able to collect data logs of particular period we can proceed to the next step.

##### *Import of data logs into RB process improvement*

Input to this step is data log conversion rules. These rules show how to identify particular activities, or whole cases in the data logs. We have to at least identify:

- Case – one pass of the process
- Activity – one step of the process
- Start time – start time of the activity

Optional data that we can extract are for example:

- Originator – originator of the particular activity
- End time – end time of the activity
- Cost – cost of the activity

We can extract large number of optional data about the process. The amount of the optional data that we can obtain depends on the readiness of the information system for process mining.

To support the comparison with the knowledge base we have to map the data from the log to the elements of the OWL. For example we map activities to object properties, some optional data we can map to entities, or data properties. It is simplified in the case of the software system that is adjusted according to our specification. In this case we have the log prepared for the automated transformation to OWL. In our framework we are based on the fact that the process model and its knowledge were made with correlation to software process knowledge framework. We assume that the logs are correctly written down with all needed information about the process, its activities and other data. So we can discover required process model and write down knowledge base in OWL. In case that we have to rely on some provider of the software system there can be a problem to communicate the requirements for structure of logs. Or there can be problem with getting the data log.

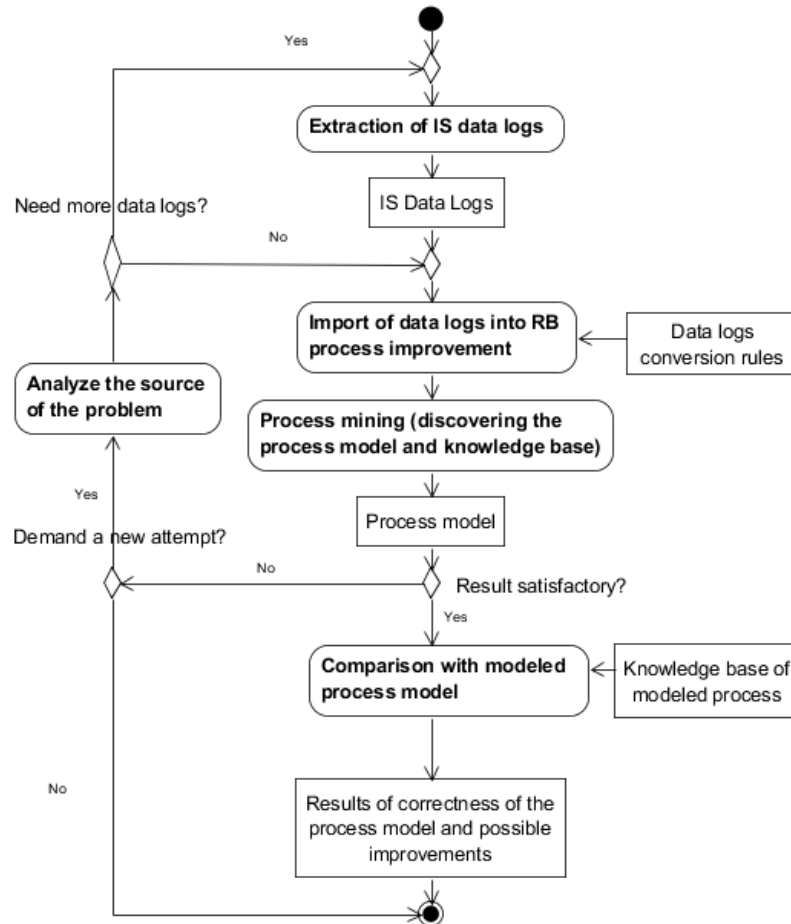


Figure 7. Check procedure – the correctness of the process model

*Process mining (discovering the process model and knowledge base)*

This is step where the model is created from the data log. We reconstruct process model from the log by BPMN Analysis (using Casual Net Miner) [2]. Result is a process model in BPMN.

Output of this step is also a knowledge base in the OWL that is based on the knowledge gathered from the log and obtained process.

*Analyze the source of the problem*

Reason of this step is to analyze where the problem is, why result – process model – is not satisfactory. Reason of this could be two types:

- We did not collect enough data logs. There is not enough data about current process in the data logs, the data are not complete or miss some mandatory fields. Then we have to try collect more data about it.

- We identified mandatory elements of the data log wrong, not complete or conversion rules are not right. We have to revise the conversion rules and identification of the data in the step *Import of data logs into RB process improvement*.

#### *Comparison with modeled process model*

This step helps us to find out if the process model that was modeled by our framework is followed correctly in the real usage.

Comparison is based on the two levels – knowledge base level and process model level. On the knowledge base level we can discover if the process has the same entities, data and object properties. For example if there are some different object properties that represent some different connections between the entities in the reality then modeled ones etc.

On the process model level we can discover the usage of particular path of the process. There we can see how often is used the main path of the process, or if there exists some activities of the process that are not used in the real enactment of this process. Analysis of most frequented paths gives us possibility to find most and least frequented paths. Then we can analyze why these paths are not frequently used, etc.

We can discover if there exist some bottlenecks that we can try to avoid in the next iteration of the process improvement. We can also find out if there exist some deviations in the process. From example if the process is not correctly followed and some mandatory activities are skipped.

#### **4. Benefits**

Before we can discuss results and benefits of the proposed methodology, we should say that similar approach of process modeling works already in practice supporting requirements management and configuring modules in healthcare information systems environment of new generation [16]. The implementation is based on embedded workflow system that allows process modeling based on building blocks generated from knowledge base of modules [15] that are implemented as configurable realizations of use cases. The idea combining this approach with formal systems, UML and enterprise framework in context of software processes led to this comprehensive research of presented methodology on universal level.

The benefits of the proposed software process modeling approach are:

- Complex software process methodology based on combination of well-known techniques – UML, OWL, Zachman Framework. Readability and ease of use of the framework are preserved by utilization of semi-formal models combined with knowledge support;
- Existence of generic software process vocabulary and rules in a knowledge base that can be easily shared, supported by community and in case of OWL based on XML also easily processed with machines;
- Graphical tools support of process building combined with ontology builders – existence of proper methodology and domain knowledge base gives us building block for process modeling in an organization;

- Support of the software process modeling with the knowledge base in the way of using the building blocks acquired from the base in the modeling tool;
- Integration of the results in the enterprise framework based on proper methodology gives us clear and transparent way of software enterprise overview;
- Approach would improve process planning, analyze and evaluate tasks quantitatively, assess costs and benefits of applying new tools and technologies on a project, train and support project staff and improve the communication with the customer;
- Integration of the models to simulation tools to simulate the progress of the processes
- Integration of the framework with the process reverse engineering (process mining) gives us complete solution for software process knowledge support and continuous process improvement. We can set up the process, implement it, execute and review the process by reverse engineering to adjust the process to be more efficient.

## 5. Conclusion and future work

In this paper, we have presented that the software process modeling using semi-formal techniques (represented by UML) could be successfully supported with knowledge bases and enterprise frameworks (with the combination of OWL and Zachman Framework). We have briefly demonstrated the modeling approach on concrete example of software process, namely the requirements management that seems to be one of most complex and critical parts of software development. Even though the state of the art of the knowledge support of business (and software) processes modeling proves us that this approach is useful and innovative, the methodology is still intuitive and hasn't been formalized yet. Big advantage of this technique is that it can be used partially even without complete formal definition because its foundations grow from well-known semi-formal approaches like UML and Zachman Framework enterprise scheme and could be used with reverse process engineering instead of conventional forward modeling approaches; thus could be used as a formal "how-to" guide for certain sub-processes of software process (e.g. presented requirements management).

The methodology needs a good balance between readability and formal definition level so it would really support the process where semi-formal techniques aren't effective and strict formal systems are too hard to use and rigid; obviously, further research and studies are required to cover all problems as the domain of software process management and requirements specification are quite complex.

Formalized adaptive approaches to requirements and quality management may be a fresh air in an unstable world of agile development and endlessly changing environments, markets and needs; full description of the procedures fitted into clear enterprise scheme may bring the software processes development to the next level.

Future work coming out of the results planned above should be focused on further development of the methodology and its utilization in process maturing and improvement that is a key element of every organization's strategy, because a more mature software process means higher quality and less expensive software products and services. If the company wants to have a more mature process, the process must follow



appropriate good practices for a higher level. One of the possible ways how achieve the improvement is to model the process [24] or simulate the model to check the improvement's goals [27]. The approach described in this paper is simulation-friendly and its models are simulation-ready as it is based on UML and formal OWL knowledge support.

### Acknowledgements

Michael A. Košinár is supported as a *Grand aided student of Municipality of Ostrava, Czech Republic*.

This research has been supported by the internal grant agency of VSB Technical University of Ostrava, Czech Republic, project no. SP2014/157 "Knowledge modeling, simulation and design of processes".

### References

- [1] Aalst, W.M.P. van der, A. Kumar (2003). *XML-based schema definition for support of interorganizational workflow*, Information Systems Research 14(1), pp. 23–46.
- [2] Aalst, W.M.P. van der, Adriansyah, A. & Van Dongen, B. 2011, *Causal nets: A modeling language tailored towards process discovery*.
- [3] Allweyer, T. (2010). *BPMN 2.0 – Introduction to the Standard for Business Process Modeling*. BoD. ISBN 978-3-8391-4985-0.
- [4] Antonionou G, Harmelen F. (2004). *A Semantic web primer*. MIT Press.
- [5] Booch, G., Jacobson, I., Rumbaugh, J. (1999). *The Unified Modeling Language User Guide*, Addison Wesley Longman, Inc..
- [6] Brooks F.P. (1987). *No Silver Bullet - Essence and Accidents of Software Engineering* (reprinted form information processing 86, 1986). Computer 20 (4): 10-19.
- [7] Curtis B., Kellner M.I., Over J. (1992). *Process modeling*. Commun ACM 35 (9): 75-90.
- [8] Dumas, M., Aalst, W.M.P. van der, Hofstede, A.H.M. ter, *Process Aware Information Systems: Bridging People and Software Through Process Technology*, Wiley- Interscience, 2005
- [9] Hug, C., Front, A., Rieu, D., Henderson-Sellers, B. (2009). *"A method to build information systems engineering process metamodels."* Journal of Systems and Software no. 82 (10):1730-1742. doi: 10.1016/j.jss.2009.05.020.
- [10] Humphrey, W.S. (1995). *A Discipline for Software Engineering*. Addison-Wesley Professional.
- [11] Jennings, N.R., Faratin, P., Norman, T.J., O'Brien, P., Odgers, B. (2000). *Autonomous agents for business process management*, International Journal of Applied Artificial Intelligence 14(2), pp. 145–189.
- [12] Kammer, P.J., Bolcer, G.A., Taylor, R.N., Hitomi, A.S., Bergman, M. (2000). *Techniques for supporting dynamic and adaptive workflow*, Computer Supported Cooperative Work 9(3–4), pp. 269–292.
- [13] Kellner, M. I., Feiler, P. H., Finkelstein, A., Katayama, T., Osterweil, L. J., Penedo, M. H., Rombach, H.D. (1990). *Software Process Modeling Example Problem*. Paper read at Software Process Workshop, 1990. 'Support for the Software Process', Proceedings of the 6th International, 28-31.
- [14] Klein, M., Dellarocas, C., (2000). *A knowledge-based approach to handling exceptions in workflow systems*, Computer Supported Cooperative Work 9(3–4), pp. 399–412.
- [15] Košinár, M. (2010a). *Design and Utilization of Knowledge Bases for Software Processes*, VŠB-TUO.
- [16] Košinár, M. (2013). *Knowledge Modeling of Agile Processes in Healthcare Systems Development*, In Proceedings of the Trendy v biomedicine, Technická univerzita v Košiciach, Slovakia.
- [17] Košinár, M., Duží, M., Kožusznik, J., Štolfa, S. (2012). *Knowledge-base approach to software-process development based on TIL*; In Proceedings of the 22nd European-Japanese Conference on Information Modeling and Knowledge Bases; Editors: Yasushi Kiyoki, Takehiro Tokuda; 2012, Prague, Czech
- [18] Košinár, M., Kožusznik, J., Štolfa, S., Duží, M., Čihalová, M. (2011). *Knowledge Support for Software Processes* in journal Evaluation of Novel Approaches to Software Engineering, Springer-Verlag.

- [19] Košinár, M., Štolfa, S., Kožusznik, J. (2010) *Knowledge Support for Software Process*. In In proceedings 5th International Conference Evaluation of Novel Approaches to Software Engineering - ENASE 2010.
- [20] Kožusznik, J., Štolfa, S. (2011). *Knowledge based approach to software development process modeling*. Paper read at Digital Information Processing and Communications, at Ostrava.
- [21] Kožusznik, J., Štolfa, S. (2011a). *Basic Constructions for the Definition of the Software Development Process with Formal Method* Paper read at The 2011 European Simulation and Modelling Conference (ESM2011), at Guimaraes.
- [22] Lenat, D., G. Miller, Yokoi, T. (1995). "CYC, WORDNET, AND EDR - CRITIQUES AND RESPONSES - DISCUSSION." *Communications of the Acm* no. 38 (11):45-48. doi: 10.1145/219717.219757.
- [23] Machado, E.P., Jr. Caetano Traina, Myrian R. B. Araujo. (2000). *Classification Abstraction: An Intrinsic Element in Database Systems*. In Proceedings of the First International Conference on Advances in Information Systems: Springer-Verlag.
- [24] Makinen, T., Varkoi, T. (2008). *Assessment driven process modeling for software process improvement*. Paper read at Management of Engineering & Technology, 2008. PICMET 2008. Portland International Conference on, 27-31 July.
- [25] Mossakowski, T., Maeder, C., Luttich, K. (2007). "The heterogeneous tool set, HETS." In Tools and Algorithms for the Construction and Analysis of Systems, Proceedings, edited by O. Grumberg and M. Huth, 519-522. Berlin: Springer-Verlag Berlin.
- [26] Narendra, N.C. (2004). *Flexible support and management of adaptive workflow processes*, Information Systems Frontiers 6(3), pp. 247–262.
- [27] Raffo D.M., Wakeland W. (2008), *Moving Up the CMMI Capability and Maturity Levels Using Simulation* (trans: Institute SE).
- [28] Raffo, D.M. (1996). *Modeling software processes quantitatively and assessing the impact of potential process changes on process performance*, Carnegie Mellon University.
- [29] Ren, A. H. (2001). *Research on the Concurrent Software Developing Method Based on Object Oriented Petri Nets*, BHU, The school of computer science, pp. 116–128.
- [30] Sawyer, P., Paech, B., Heymans, P. (2007). *Requirements Engineering: Foundation for Software Quality*. page 191.
- [31] Scacchi, W. (1999). "Experience with software process simulation and modeling." *Journal of Systems and Software* no. 46 (2-3):183-192.
- [32] Singh, M.P., Huhns, M.N. (2005). *Service-oriented computing: semantics, processes, agents*. London: Wiley.
- [33] Smith, John Miles, and Diane C. P. Smith. (1977). "Database abstractions: aggregation and generalization." *ACM Transactions on Database Systems* no. 2 (2):105-133. doi: 10.1145/320544.320546.
- [34] Štolfa S., Kožusznik J., Košinár M., Duží M., Čihalová M., Vondrák I. (2010). *Building Process Definition with Ontology Background*. Paper presented at the CISIM 2010, Krakow, Poland.
- [35] Štolfa, J., Kopka, M., Štolfa, S., Koběorský, O., Snášel, V., An Application of Process Mining to Invoice Verification Process in SAP, *Innovations in Bio-inspired Computing and Applications*, 2014, 61-74
- [36] Vergidis K., Tiwari A., Majeed B. (2008). *Business Process Analysis and Optimization: Beyond Reengineering*. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 38 (1): 69-82.
- [37] Vondrák, I. (2005). *Methods for Software Specification*, VŠB - TU Ostrava.
- [38] Wang, T.D., Parsia, B.; Hendler, J. (2006). "A Survey of the Web Ontology Landscape". *The Semantic Web - ISWC 2006. Lecture Notes in Computer Science* 4273. p. 682.
- [39] Weske, M., Aalst, W.M.P. van der, Verbeek, H.M.W.E., *Advances in business process management*, Data & Knowledge Engineering 50 (1) (2004) 1–8.
- [40] Workflow Management Coalition (1999). *Workflow Management Coalition Terminology & Glossary* (Document No. WFMC-TC-1011). Workflow Management Coalition Specification.

# A Metadata System for Quality Management

Frank KRAMER<sup>a</sup> Bernhard THALHEIM<sup>a</sup>

<sup>a</sup> *Information System Engineering, Christian-Albrechts-University at Kiel, Germany*

**Abstract.** Data quality becomes an important issue for scientific databases. It is currently managed insufficiently. General solutions for managing the quality data on the object level are represented insufficiently nowadays. These applications need, however, a management system that supports quality treatment over the entire lifespan. We develop an approach to quality management that can be integrated into a metadata management system. This approach is based on a separation of concern through database components, extended views and Information Containers.

**Keywords.** Metadata Management, Data Quality Management, Conceptual Modelling, Component-based Development, Content Types, Information Container

## 1. Introduction

One of the challenges imposed by science databases besides their volume, velocity, variety and veracity is the characterisation of these data in such way that any collaborating partner can use this data in the best or most appropriate form: This information is typically given by means of metadata. The *Deutsche Forschungsgesellschaft* published a guideline with seven main tasks for good long term research data management [5]. One of these tasks is data quality characterisation. Data quality has already been an issue for classical database systems, e.g. [1], [18], [12] or [14]. Most of the problems are, however, still unsolved for data quality management. Moreover, there are very few tools currently available, e.g. in 2006 the work of [1] identifies only six tools for data quality management. Since quality characterisation is constantly under change and a large variety of quality characterisation parameters must be considered, the development of quality models results in large and evolving schemata. Therefore, we need a technology that supports a flexible and simple quality management.

We observe however, that quality characterisation can be based on different dimensions such as data characterisation, business event characterisation, business deploymentability, stakeholder involvement, provenance, privacy and security, reliability and trustworthiness for imported or raw data etc. These dimensions allow a separation into different concerns based on a component technology. Al-

ready, classical database management has been deploying components for human resources, product, production, trade, and finance services. A component is *a database schema that has an import and an export interface by which it may be connected to other components via a standardized interface technique* [22]. Component technology reduces development, allows reviews of solutions already developed and simple replacement of a component by an improved version. This technology can also be used for metadata. Since metadata may also be considered as a dimension of database schemata. Therefore we propose in the following a component approach to metadata management and illustrate this approach for quality metadata.

The interaction among components cannot be based on simple relational one-table views. Therefore, we develop Content Types that represent a whole interaction subschema. These Content Types generalize view technology. This extension allows to include also service functions for utilization of the database objects. Information Containers are used to deliver data together with functions to federated components. These Information Containers are based on the theory of Tuple Spaces. Tuple Spaces support a flexible exchange of data in varying formats and compositions.

Therefore, we will first introduce Tuple Spaces, Content Types, Information Container, database components and a wiring technology for the composition of components in Section 2. Section 3 uses component technology for metadata management. We can separate metadata management into a component layer, content object layer, container layer and interface layer. In Section 4, we refine this approach to a quality management. This quality management is conceptualised by a separate database structure and is underpinned by corresponding content objects. Finally, Section 5 surveys related work.

## 2. Basic Concepts

This Section presents the general concepts of *Tuple Spaces*, *Content Types*, *Information Container* and database components. These build the basis for the construction of the metadata management system that is shown in section 3.

### 2.1. Tuple Space

The concept of *Tuple Spaces* was first introduced as distributed data structure for the coordination language Linda in the 1980's at Yale University. This Section gives a short introduction into *Tuple Spaces* from [3], [2], [7] and [13].

A *Tuple Space* is a distributed data structure that can be manipulated by an arbitrary size of parallel processes. It contains a bag of data objects called tuple. A tuple is a sequence of fields. Every field in a tuple can be an expression, a value or a multi-typed variable. Selecting tuple from the *Tupel Space* is realized with pattern matching. Two tuple from the *Tuple Space* match, if they have the same

quantity of fields and the related fields match in their variables or their values. A variable matches only values with the same type and a value only matches an equal value. In the basic concept a process has only four operations to manipulate the *Tuple Space*. *Out(t)* inserts a tuple into the *Tuple Space*. With *eval(Q)*, a process Q is added to the *Tuple Space*, e.g. the evaluation of a special value for another process. After termination of Q, the process turns into a tuple in the *Tuple Space*. These two operations are non blocking, i.e. a process performs *out(t)* or *eval(Q)* and continues with its task. A tuple from the *Tuple Space* can be read by performing *read(t)*. Three possible cases can arise. First, only one tuple matches t. Then, this tuple is selected from the *Tuple Space*. Second, if more than one tuple matches t, the first matching tuple is chosen non-deterministically. Third, no tuple in the *Tuple Space* matches t. In this case, the process that performs the read must wait until a matching tuple can be found. So *read(t)* is a blocking operation. If the tuple can not only be read, but also be deleted from the *Tuple Space* a process can perform an *in(t)*. The matching behaviour is the same as by *read(t)*.

## 2.2. Content Types and Information Container

The concepts of *Content Types* and *Information Container* were first introduced in [4], in the context of *Information Units* and *Information Container* for *Information Services*. Later, *Information Units* were refined in [19] as *Content Types* for information systems. An *Information Service* is database-backed, i.e. it accesses to a database that contains all structured data the service needs. Hence, a very important task for the construction of the *Information Service* is the conceptual modelling of the database design for the service. This includes not only the modelling of the structural database schema but also the design of static and dynamic integrity constraints, database processes and user interfaces. Therefore, the conceptual modelling can be divided into the two dimensions. One dimension covers the global and local aspects and the other dimension covers static and dynamic aspects.

A *Content Type* is a model of a local static component of an information service. It represents generalized views on the global static database schema for the whole information service. This allows a good representation of the data for the user of the system. The basis for a *Content Type* builds a database view V that is generally defined for databases in listing 1.

```

CREATE VIEW name (projection variables) AS
SELECT projection
FROM database schema
WHERE condition
GROUP BY expression for grouping
HAVING selection among groups
ORDER BY sorting criteria

```

Listing 1: basic SQL view

Such a view generates a logical view over a database schema but it cannot be used to describe conceptual related objects for a process within a system. Hence a structure is needed that allows to specify a set of correlated views for different tasks with a defined functionality for the views and a customer adjusted representation of the data within the views. Consequently, a general view is defined based on [20] in listing 2.

```

GENERATE mapping: vars → output structure
FROM database types
WHERE selection condition
REPRESENT USING general representation style
    &abstraction(granularity , measure , precision)
    &orders within the representation
    &points of view
    &hierarchical representation
    &separation
BROWSING REPRESENTATION condition
    &navigation
FUNCTIONS search function
    &export function
    &import function
    &manipulation function
    &session function
    &marking function

```

Listing 2: generalised SQL view

This generalized view represents the content type within a system. An instantiation of such a type is called *Content Object*. A *Content Object* is allocated to the user of the system. The data within such an object can be classified into retrieval, input, output, display and collateral data. Retrieval data is taken from the database and used within the dialogue steps. Input data covers the data that is put into the content object from the user. Output data is inserted into the underlying database system through the content object. Display data is shown to the user to give better information within his working process. Collateral data summarizes data from already arranged tasks within the process.

To generate a content type, a rule-based system is deployed. It consists of three types of rules which are applied successively to the global schema to get the content type. The first type of rules are *computational rules* that can be used to filter the local view schema from the global schema. For instance, a subschema of the global schema is a result of applying computational rules on the global schema. The second type of rules are *abstraction and rebuilding rules*, which are used to summarize. The output is a *raw content type* that consists of the abstraction and construction of pre-information from the result of the computational rules. *Scaling rules* are the last type of rules in the system. With the help of information

about the user of the service, such as a user profile and interests, the information get customized in its representation. A rule system with these three types of rules is the smallest, possible system that exists for this task. As a consequence of being an open rule system it can be extended with other systems, for example, an analysis system, to allow the analysis of data sets within the *Information Unit*.

The dynamic, global component of an information service is called a process. This process consists of a set of local dynamic components called dialogues. Again, the dialogues contain a set of dialogue steps a user can perform. However the dialogues depend not only on the process. Each dialogue needs a collection of *Content Types* which provides a dialogue with the input information of interest. This collection of *Content Types* is called an *Information Container*. An *Information Container* adapts the concept of *Tuple Spaces* which is described in section 2.1. Hence, an *Information Container* for dialogues can be defined as  $IC = (TS, Cap, Load, Unload, Style, Escort)$  with the following parameters:

- *TS*: The *Tuple Space* the container is made of. This *Tuple Space* covers the content objects for the process.
- *Cap*: The capacity of the container which shows the allowed size and content types within its tuple space.
- *Load*: The loadability of the container that parametrizes the computational functionality for inserting content objects into the container. An example for such functionality is, to prefetch information into the container.
- *Unload*: The unloadability of the container which defines the ability to read, scan and survey content objects from the container.
- *Style*: Rules to define the layout of the container. They are based on user profiles and their preferences, expectations and environment.
- *Escort*: Information that depends on the instantiation of the container. It is used to guide a user from a current state of a dialogue step to the next possible step. Furthermore it contains special background information for the user.

The instantiation of an *Information Container* depends on the rules and the supported function of the collected *Content Type*. *Tuple Space* functionality is used to load the provided information into the container.

### 2.3. Database Components and Harnesses

The design of database components is described in [15] and [22]. The construction of a component is based on the Higher-Order Entity-Relationship Model (HERM) [21]. In HERM a database type is defined as  $\mathfrak{S} = (\text{Struc}, \text{Op}, \Sigma)$  with a structure *Struc*, a set of operations *Op* and a set of static integrity constraints  $\Sigma$ . The structure is defined by a recursive type equality  $t = B|t \times \dots \times t ||\{t\}|:t$  over a set of basic data types *B*, a set of labels *L* and constructors for tuple (product), set and bag. A database schema  $S = (\mathfrak{S}_1, \dots, \mathfrak{S}_m, \Sigma_G)$  is given by a set of database types  $\mathfrak{S}_1, \dots, \mathfrak{S}_m$  and a set of global integrity constraints  $\Sigma_G$ .

Formally, a component can be described as input-output machine. Every machine gets a set of all database states  $S^C$ , a set of input views  $I^{\mathfrak{W}}$  and a set of output

views  $O^{\mathfrak{V}}$ . A view can be defined as  $\mathfrak{V} = (V, Op_V)$  with an algebraic expression  $V$  on a database schema  $S$  and a set of HERM algebra operations  $Op_V$  on the view  $V$ . The views are used for the collaboration of the components by exchanging data over them. Therefore, an input view of one machine can be connected to an output view of another machine. This data exchange is done by a channel  $C$ . The structure of the *channel* is defined by a function  $type : C \rightarrow \mathfrak{V}$  that maps a channel  $C$  on a corresponding view schema  $V$ .

A database component is defined as  $\mathfrak{K} = (S_{\mathfrak{K}}, I_{\mathfrak{K}}^{\mathfrak{V}}, O_{\mathfrak{K}}^{\mathfrak{V}}, S_{\mathfrak{K}}^C, \Delta_{\mathfrak{K}})$  with a database schema  $S_{\mathfrak{K}}$ , input and output views  $I_{\mathfrak{K}}^{\mathfrak{V}}$  and  $O_{\mathfrak{K}}^{\mathfrak{V}}$ , all states of the database  $S_{\mathfrak{K}}^C$  and a channel function of the input and output views  $\Delta_{\mathfrak{K}} : (S_{\mathfrak{K}}^C \times (O_{\mathfrak{K}}^{\mathfrak{V}} \rightarrow M^*)) \rightarrow \mathfrak{P}(S_{\mathfrak{K}}^C \times (I_{\mathfrak{K}}^{\mathfrak{V}} \rightarrow M^*))$  with a set of words  $M^*$  of the underlying database structure. To connect two components together, they must be free of name conflicts and the input and output views have to be domain-compatible. Assume two components  $\mathfrak{K}_1 = (S_1, I_1^{\mathfrak{V}}, O_1^{\mathfrak{V}}, S_1^C, \Delta_1)$  and  $\mathfrak{K}_2 = (S_2, I_2^{\mathfrak{V}}, O_2^{\mathfrak{V}}, S_2^C, \Delta_2)$ . They are free of name conflicts, if the names of their entity, relationship and attribute names within their schema  $S_1$  and  $S_2$  are disjoint. Two channels  $C_1$  from  $\mathfrak{K}_1$  and  $C_2$  from  $\mathfrak{K}_2$  are domain-compatible, if  $dom(type(C_1)) = dom(type(C_2))$ . So the output  $O_1^{\mathfrak{V}} \in O_1^{\mathfrak{V}}$  of component  $\mathfrak{K}_1$  is domain-compatible to input  $I_2^{\mathfrak{V}} \in I_2^{\mathfrak{V}}$  of component  $\mathfrak{K}_2$  when  $dom(type(O_1^{\mathfrak{V}})) \subseteq dom(type(I_2^{\mathfrak{V}}))$ . For the definition of unification, permutation and renaming of channels together with the introduction of fictitious channels and the parallel composition of channels with feedback we refer back to [22].

The modularisation of a database schema with components is then used, to scale the schema into dimensions. With the dimensioning of a schema, it is possible, to store data, based on their origin and purpose. This reduces drastically the complexity of the whole database schema. In consequence of the data dimensioning, only small components exist and not a huge, global schema. To connect components that lie in different dimensions, we will use a concept called harness. Harnesses are described in [15] and [22]. The behaviour of a harness is similar to the behaviour of wired harnesses in electrical engineering. Formally, a harness is based on a harness skeleton. This is a special form of metaschema architecture. The skeleton consists of a set of components and a set of harnesses that represent the overlapping functions of the components. A  $n$ -ary harness skeleton can be defined as a triple  $\mathfrak{H} = (\mathcal{K}, \mathcal{L}, \tau)$  with a set of components  $\mathcal{K} = \{\mathfrak{K}_1, \dots, \mathfrak{K}_m\}$ , a set of labels  $\mathcal{L} = \{L_1, \dots, L_n\}$  having  $n \geq m$  that represent roles of components in the skeleton and a total function  $\tau : \mathcal{L} \rightarrow \mathfrak{K}$  that assigns a component to its roles. Therefore, a harness is defined as  $\mathcal{H} = (\mathcal{K}, \mathcal{L}, \iota \circ \tau)$ , composed of a harness skeleton  $(\mathcal{K}, \mathcal{L}, \tau)$  and a filter function  $\iota(L_i) = l$ . A filter connects a view from the component  $\mathfrak{K}_j$  with a label  $L_i$ , if  $\iota(L_i) = l$  for  $j = \tau(L_i)$  and  $l \in \{V_1^{\mathfrak{K}_j}, \dots, V_{l_{\mathfrak{K}_j}}^{\mathfrak{K}_j}\}$  holds.

### 3. A Component Based Metadata Management System

This part of the paper presents a component-based metadata management system. It can be integrated into every existing database backing application. The



only condition for the database is a component-based construction that is described in Section 2.3. The metadata that is managed in this system is also stored as components. This generates a compact management of the metadata in the system without using extended views. Furthermore, we get a database independent system because our model can be connected to every component structure, if import and export views are available.

### 3.1. An Independent Metadata Management System

For our approach of metadata management we will use a four layer metadata management system as shown in figure 1. It is inspired by the work of [10] for a generic database V-Architecture Model.

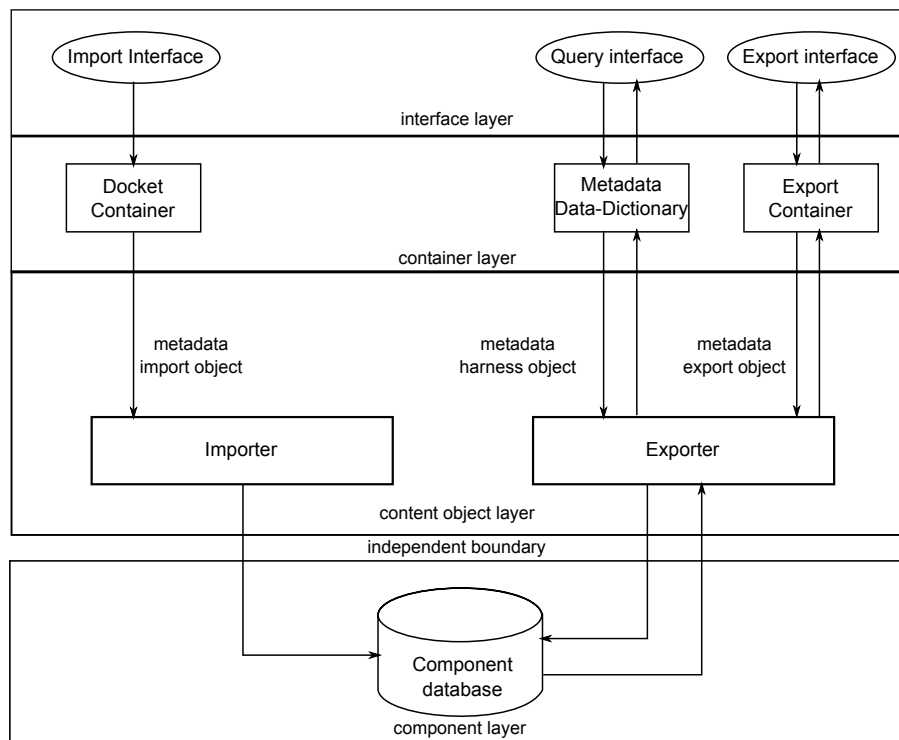


Figure 1. Four layered Metadata Management System

The component-based database builds the bottom layer of our system. The second layer of our model is the *Content Object Layer* for managing the import and export of data. An *Importer* is used to send every create, update and delete ( CRUD ) operation into the corresponding database component. For this, the importer creates *Content Types* for data import over the input view of a database component. If a CRUD operation is performed over a *Content Type*, its export functions are responsible for the import into the right input view of the corresponding component. If data should be read or exported from the database an *Exporter* is used to get the data from it. Like the *Importer*, the *Exporter* creates *Content*

*Types*. But the *Content Types* from the *Exporter* represent the output views from the database components. To generate the *Content Type*, the *Exporter* selects the requested data from the output views of the corresponding components and creates the *Content Object*. Section 3.2 will take a closer look on three *Content Types* that are needed.

Above the *Content Object Layer* lies a *Container Layer*. The system consists of *Information Container* similar to the container presented in section 2.2. A container builds the workspace for a user of the system. There are three types of *Information Container* in this layer. Every container gets an individual style parameter which is customized exactly for the user's needs, and an escort parameter that guides the user through the usage of the container. The first type is the *Docket Container*. It is used to import metadata into the system. This import process is based on docket. As described in [16], docket is used to review data from a content provider. If a user wants to import metadata for an existing value to the system, a defined docket is created from a corresponding *Docket Container*. The second *Information Container* is the *Metadata Data-Dictionary Container*. A *Data-Dictionary Container* is instantiated from *Content Objects* of the *Exporter*. A container builds the Metadata Data-Dictionary by connecting output views of metadata components and the relevant output views of application and workflow components through its *Content Objects*. The third container, is the *Export Container*. All objects that can be exported to an external system can be found there. Section 3.3 will take a closer look on the information container.

To interact with the container and the objects within it, an *Import*, *Query* and *Export Interface* can be generated automatically for a user. Over the *Import Interface* a user can create and interact with the docket which are relevant for him. In these components the user can import data over the functionality which is defined with the load parameter of the container. To get data from the Data-Dictionary, and work with the metadata, the *Query Interface* can be used. Over the interface, a user can perform all functions which are defined in the unload parameter of the component, to find the metadata information he needs. To export specific objects to another system or document, the *Export Interface* can be used. With this interface a user can determine which metadata should be exported and how this export is constructed.

### 3.2. Metadata Content Types

This Section will present the three types of metadata content types and take a look on the needed functionality. The first type is the *Metadata Import Type*. It is generated over the input views  $I_{\mathfrak{R}}^{\mathfrak{M}}$  of a metadata database component  $\mathfrak{R} = (S_{\mathfrak{R}}, I_{\mathfrak{R}}^{\mathfrak{M}}, O_{\mathfrak{R}}^{\mathfrak{M}}, S_{\mathfrak{R}}^C, \Delta_{\mathfrak{R}})$ . Every import type can allocate a set of functions representing the functionality that can be applied to the data within an instance of the type. With *search functions* a user can look for specific metadata types he wants to import into the system. With a *value search*, a user can look for specific metadata within a metadata component, for example, the longitude of a measurement. Moreover a user can *zoom out* or *zoom in* into a metadata component to generalize or spe-

cialize the import range he wants to access. To order the sequence of data within the type, a *reorder* functionality can be implemented. *Navigation* functions allow a user to go through the metadata component and import values in a controlled way. If a user wants to see what kind of other metadata types are associated in the context of the regarded object, a *Content Type* provides *contextual* functions that compute such dependencies. As an example take a special quality parameter such as correctness. A contextual function can then give all other quality parameters that are contrary to this parameter. To get an overview over the data within the media object, *review* functions can exist.

The core functionality of an importer is *data manipulation* and *data import*. There can be two types of data manipulation within a type. First, a user can manipulate the data in the database component with a *database manipulation function*. Then, the functions of the import type work directly on the import views of the database component from which the type is instantiated. Furthermore, a user can only manipulate the data within the import type with *object manipulation* functions. This is useful for temporary changes to the metadata. For example, take metadata that are only manipulated for a needed interim stage but do not appear in the database. For the import of data the type must have the functionality to import data from a user input or from external data sources such as files through *data import functions*. Further, there has to be a functionality to import data from one import object to another import object of the same type by *object import functions*. User interactions before an instantiation of the import object must also be intercepted from *interaction functions* of the import type. Import types have only importing tasks. However, there are two special *export* features of import types. To load an object of a special type from the database in the workspace of a user, *integration functions* are needed. These functions are called from the information container. Information container will be described in 3.3. Another functionality is the option to pass an import type to another user by *pass functions*. So, it is possible for a user to import his metadata, and then export it to another user that can extend the same object with other metadata without a new instantiation of the object. Also, the temporary changes can be used for other import steps. For the same reason the import type contains *mark* functionality. A user can use *comment* or *colour functions* to mark special metadata within an object that is important for another user who gets this special import object. The last part of functionality is the *session management function* of an import type. Every import type gets functionality to *open*, *log* and *close* an object, plus a *restore function* if an error occurs during data import. In most cases, this is simply solved by cookies as found in web business.

The second *Content Type* is the *Metadata Harness Type*. It connects an application data output view to a combination of the output views from the metadata components in the database. Harnesses are used in our metadata management system to connect the output views of application and metadata components. A user can get information about the metadata that is assigned to the examined application data. As a result of connecting only output views a harness type has read-only functionality and no export functionality except the integration func-

tions for container load. Furthermore, there are also no import and data manipulation functions for this type. If a user wants to insert data, he must use the corresponding import type. Similar to the import types a harness type has *search* functions for the data within the type. Search functionality is the main function of the harness type as a user can explore metadata with the type and searching is the main functionality for exploring data. As for the import type, there exist functions for value-search, zooming, reorder, navigation, getting contextual and review information. This functionality is enhanced with *association* functions. These functions make it possible to get associated connections between different data across different objects of the same type. With these functions complex objects can be build that contain associated objects. To work with the metadata, the *mark* functionality is also important for the harness type. With these functions a user can color or comment important metadata information he gets from the Data-Dictionary. Like import types harness types get *session* functionality too.

To export data from the metadata management system into an external source, for example, a data centre or another database, the *Metadata Export Type*, can be used. It connects an application data output view  $O_{\mathfrak{D}}^{\mathfrak{D}}$  of a component  $\mathfrak{D} = (S_{\mathfrak{D}}, I_{\mathfrak{D}}^{\mathfrak{D}}, O_{\mathfrak{D}}^{\mathfrak{D}}, S_{\mathfrak{D}}^C, \Delta_{\mathfrak{D}})$  with a combination of the output views  $O_{\mathfrak{R}_i}^{\mathfrak{D}}$  from the metadata components  $\mathfrak{R}_i = (S_{\mathfrak{R}_i}, I_{\mathfrak{R}_i}^{\mathfrak{D}}, O_{\mathfrak{R}_i}^{\mathfrak{D}}, S_{\mathfrak{R}_i}^C, \Delta_{\mathfrak{R}_i})$  ( $1 \leq i \leq 6$ ). Due to the fact that an export type only connects output views an export type has read-only functionality like a harness type. As a result of this, there is no import functionality for this type. Only one of the data manipulation function exist. Data within an export type can be manipulated with object manipulation functions. With this functionality, data can be manually adapted, if it does not match the rules of the external source. For example, the longitudes and latitudes of research data should be exported to a data centre. If a user recognizes that one pair of this data is transposed in the export type, he can correct it for the export without reconstructing why this error occurs. For changing the data in the database, only the import type can be used. The main functionality for this type is the *export* functionality. The export functionality not only covers the functionality from the import type, furthermore, there are functions for export and integration of data into documents for the export into external sources. With *document export functions* viewed metadata can be exported to an external document style like CSV or XST so that this data can be printed for external use or imported into other systems. With *data integration functions*, examined metadata can be integrated into other documents that already contain other data for the underlying application data. As an example take a XML document that contains the workflow information for application data, and the metadata must be integrated into the document. The main difference between document export and document integration functions is the fact that the first class of functions create new documents while the integration functions extend existing documents. Like import types, export types get *mark* and *session* functionality. Unlike import and harness types, there is no search functionality within this type. If a user wants to investigate the data, he can use the harness type.

### 3.3. Metadata Information Container

Within the metadata management system a user normally does not only want one import and one export object. Therefore, the metadata management system provides a workspace for the user where he can work on a collection of different import, query or export objects. Every object allocates the functionality that is defined for the type. The workspace is defined by an *Information Container* based on the description in Section 2.2 of this paper. An *Information Container* is defined as an abstract state machine  $\mathfrak{C} = (\mathcal{J}, \mathcal{M}, \mathcal{O}, ops_{\mathfrak{C}}, \Sigma_{\mathfrak{C}})$ .  $\mathcal{J}$ ,  $\mathcal{M}$ ,  $\mathcal{O}$  are *Tuple Spaces* as described in section 2.1. Every element within the *Tuple Space* consists of pattern (*Key, Content Object*). The *Tuple Space* has a bag structure, because it can contain the same object multiple times. To identify elements within the *Tuple Space*, an intelligent pattern matching algorithm must be implemented that make it possible, to identify multiple equal objects. A container covers three different *Tuple Space*.  $\mathcal{J}$  is the *Input Space*. It is used to load *Content Objects* into the container from the corresponding interface. The *Content Space*  $\mathcal{M}$  covers all extended *Content Objects*. Only objects in this space can be select from a user to work with. The extension of the object includes optional descriptions and user information about the objects within the container.  $\mathcal{O}$  is the *Output Space* of the container. It contains the requested objects from the container for user interaction. *Operations*  $ops_{\mathfrak{C}}$  are used to support the management of the state spaces. This includes operations for the import of *Content Objects* into  $\mathcal{J}$  and enhances objects in  $\mathcal{M}$ , operations for changing the state of the container and operations for the export of *Content Objects* to  $\mathcal{O}$ .  $\Sigma_{\mathfrak{C}}$  describes *limitations* of the container. This covers the capacity of the container, the cardinality of the pattern matching function and restriction for container unload.

The basic *Information Container*  $\mathfrak{C}$  can be divided into three types for import, query and export and extended with special style and escort information. A *Docket Container* is used for the import and is defined as  $DC = (\mathfrak{C}, Docket, Escort)$ . The container  $\mathfrak{C}$  is bound to the *Import Interface* of the metadata management system. The only objects within this container are import objects. The *Docket Information* is the style information that holds for objects within this container. It is based on the ideas in [16] for a docket based review processes. The import of metadata within the system is based on defined dockets a user of the system must use. To present user specified presentation options, the docket style information is container exclusive. For example, assuming that quality metadata for a research application date has to be inserted. The head of the research group may have much more quality metadata he can insert into the database, as a research group member. Thus, the head of the research group gets a docket in his container that enables much more interaction with objects that cover quality informations than the docket for a research group member. The import type that is loaded in the *Docket Container* can be the same for the head and the research group. For the same reason, every container gets exclusive escort information. *Escort* functions hold support information for the user of the container. In the example given above, the head of the research group needs more information about what quality metadata are expected than a research group member .

A *Metadata Data-Dictionary Container* is defined as  $MDD = (\mathfrak{C}, DictionaryStyle, Escort)$ . The container  $\mathfrak{C}$  is bound to the *Query Interface* of the metadata management system. So the only objects within this container are harness objects. As mentioned in 3.2 the harness objects build harnesses between application and metadata. A collection of such harness objects forms the Metadata Data-Dictionary. The *DictionaryStyle* describes the structure of the Data-Dictionary. Hence, every user can get a personal styled Data-Dictionary with an appropriated design. For example, take the head of the research group from above. The head of the research group can get a personalized Data-Dictionary where all metadata is covered for his special needs, such as administrative metadata that a research group member is not allowed to see. Similar to docket container there are also specialized escort functions that contain special user support information.

An *export container* is defined as  $EC = (\mathfrak{C}, SourceStyle, Escort)$ . The container  $\mathfrak{C}$  is bound to the *Export Interface* of the metadata management system. Thus, the only objects within this container are export objects. The container can be used to export metadata within this container to an external source. As a result, every external source gets a container. The *SourceStyle* describes how the structure of the data looks for a special source. For example, take an external data centre that defines a special rule system for the data that can be imported. A user can import objects into his *Export Container*. The *SourceStyle* represents the export rules which describe, how the data from the imported objects are transformed for the data centre. Therefore, an export object can be loaded into different containers that export this object in different sources with different styles. The escort information contains information about this structure for the user. For example, the definition of the type that is allowed for a special kind of data can be given by the escort information.

All these different containers have the same operations they perform in the *Tuple Space* for changing the state. The original *Tuple Space* model in Section 2.1 has four basic operations  $out(t)$ ,  $eval(t)$ ,  $in(t)$  and  $read(t)$ . Departing from this, a container holds three basic operations for the *Tuple Space*.  $Eval(t)$  instantiates a tuple  $t$  within a *Tuple Space*. This is similar to the  $eval(t)$  function of the original *Tuple Space*. After instantiation, the tuple can be found in the *Tuple Space* where  $eval(t)$  is performed. If the capacity of the container is exceeded by the tuple of  $eval(t)$ , the tuple will not be found in the *Tuple Space*. Because of running in parallel, a function  $success(eval(t))$  can be performed that informs the process that has performed the  $eval(t)$  that the  $eval(t)$  is done. Note that it only reports the end of  $eval(t)$  not that the capacity was exceeded and the tuple was not created. The second operation is  $find(\mathfrak{C}, m, t)$ . This operation finds all tuple within the *Tuple Space* of  $\mathfrak{C}$  that match  $t$  with the restriction of the pattern  $m$ . The last operation is  $choose(M)$  which selects an element of the *Tuple Space*. With these three operations the state transitions can be defined for the operations  $out(t)$ ,  $read(t)$  and  $in(t)$  that can be found in the original *Tuple Space* concept. Let for this  $Z = (J, \mathcal{M}, \emptyset)$  represent the states of the container,  $t \in Tuple_{\mathfrak{C}}$  represent a tuple of container  $\mathfrak{C}$  and  $m \in Pattern$  are a pattern to search for a tuple. To load a tuple within the container the function  $load : Z \times Tuple_{\mathfrak{C}} \rightarrow Z$

with  $load((\mathcal{J}, \mathcal{M}, \mathcal{O}), t) = (\mathcal{J}, \mathcal{M} \cup \{|eval(t)|\}, \mathcal{O})$  is performed. If the system should ensure that the  $eval(t)$  is successfully finished, a  $successLoad : Z \times Tuple_{\mathcal{E}} \rightarrow Z$  with  $successLoad((\mathcal{J}, \mathcal{M}, \mathcal{O}), t) = success(eval(t)) \rightarrow (\mathcal{J}, \mathcal{M} \cup \{|eval(t)|\}, \mathcal{O})$  can be executed. To read data from the *Tuple Space*, the function  $read : Z \times Pattern \times Tuple_{\mathcal{E}} \rightarrow \mathcal{O}$  with  $read((\mathcal{J}, \mathcal{M}, \mathcal{O}), m, t) = choose(find((\mathcal{J}, \mathcal{M}, \mathcal{O}), m, t))$  can be used to get a tuple  $t$  that matches pattern  $m$ . To read and remove a tuple from the *Tuple Space*, the function  $in : Z \times Pattern \times Tuple_{\mathcal{E}} \rightarrow \mathcal{M} \times \mathcal{O}$  with  $in((\mathcal{J}, \mathcal{M}, \mathcal{O}), m, t) = (set\ x = read((\mathcal{J}, \mathcal{M}, \mathcal{O}), m, t)\ return\ (x, \mathcal{M} \setminus \{|x|\}))$  must be used. With these functions the original operations on the *Tuple Space* are mapped to the abstract state machine.

#### 4. Quality Component

This part of the paper will show how the quality component for the metadata management system from Section 3 can be created. Therefore, it presents a conceptual HERM model for the quality component and shows the construction of an import and export type for quality management.

##### 4.1. A Quality HERM-Schema

The quality component is based on the *Quality Characteristics Conceptual Model* (QCCM) as found in [9]. It describes how quality data can be saved for a data object. The quality component is based on dimensions. A dimension is described with different characteristics. A characteristic consists of a set of attributes that has a metric. Figure 2 shows the design of the component.

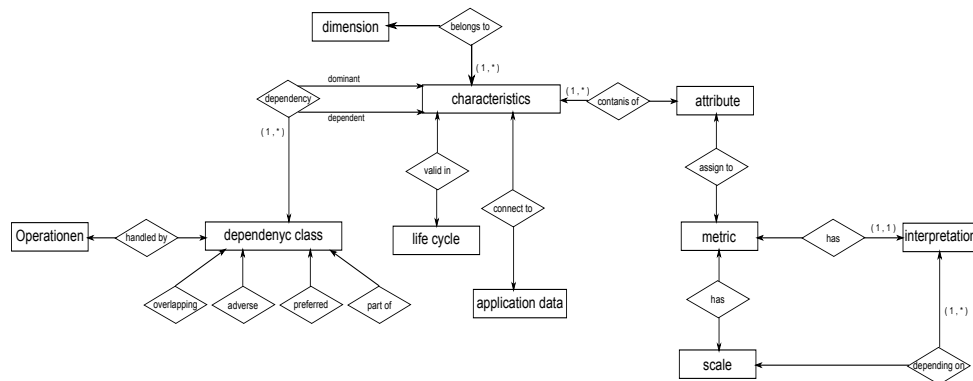


Figure 2. Quality component as HERM Schema

To point out the fact that quality metadata are connected to application data, a dummy type *application data* is introduced. It can be expressed as a connection to an export view of an application data component. This ensures that the schema is understandable and manageable in the practical usage. Hence, every application data component can get a set of quality characteristics that defines the quality of the application data. Every characteristic gets at least one dimen-

sion it belongs to. There are a lot of different proposals for quality dimensions, e.g. [1] and [18]. In Section 5 we will take a short look on different conceptual models. Furthermore, a dimension can cover user defined categories that are only relevant for the business of the users. Every characteristic gets a set of attributes and the attributes get metrics. Every attribute has a scale and an interpretation of the metric. The interpretation is distinct for each metric and depends on the minimum of one scale. As a consequence, this part of the model is capable of covering different conceptual quality models.

One dependency of one quality characteristic to another characteristic can be defined. This is due to the fact, that quality parameters are not independent in practice. Every characteristic can depend on one or more characteristics. This dependency can be assigned to one of four dependency classes. To explain these classes, assume two characteristics  $C_1$  and  $C_2$  with attribute sets  $A_1$  and  $A_2$ . If  $C_1$  gets an *overlapping* characteristic  $C_2$ , then  $A_1 \cap A_2 \neq \emptyset$ . When  $C_1$  and  $C_2$  are *contradictory* characteristics, then either  $C_1$  or  $C_2$  can be a valid quality characteristic for the application data.  $C_1$  is *in favour* to  $C_2$  when, in every quality assertion where  $C_1$  and  $C_2$  can be consulted,  $C_1$  is taken. The characteristic  $C_2$  is *part of*  $C_1$ , if  $A_2 \subseteq A_1$ . For each of these dependency classes there are operations which make sure that these dependencies are used for the quality characteristics.

#### 4.2. Quality Content Objects

The schema from the previous Section will now be used to construct content objects that can be used to import and export quality metadata to the system. In this Section, an import type for a fictitious application will be created. It will show an intrinsic quality dimension that is used to describe the quality of data within the whole system as seen in [23]. The system is used to manage scientific research data. Other arbitrary types can be created in the same way as shown here for the example application. Also, the representation of the import type is ignored as it depends heavily on an implemented application.

Assuming that there is a research group. This group has the task to insert independent intrinsic data quality attributes of research data into a database. They insert these data in the analysis phase of the research data. All quality data that is assigned to the research data before this point in the life cycle are not valid any more. To insert the data, there is a defined docket that allows a structured import of quality parameters for the members of the research group. Furthermore, there already exist a database component  $\mathfrak{R} = (S_{\mathfrak{R}}, I_{\mathfrak{R}}^{\mathfrak{Q}}, O_{\mathfrak{R}}^{\mathfrak{Q}}, S_{\mathfrak{R}}^C, \Delta_{\mathfrak{R}})$  based on the quality schema of Section 4.1 with the corresponding import views of database components. Now the import type can be build. Hence, a view schema is created that forms the basis for the import type. Figure 3 shows a possible schema.

The figure displays the left side of the basic schema. Dependencies are not needed as the independence of the parameter is assumed. A member of the research group can import any kind of intrinsic quality parameters (characteristic or attribute) into the system. To assure that only intrinsic data quality is imported, the di-



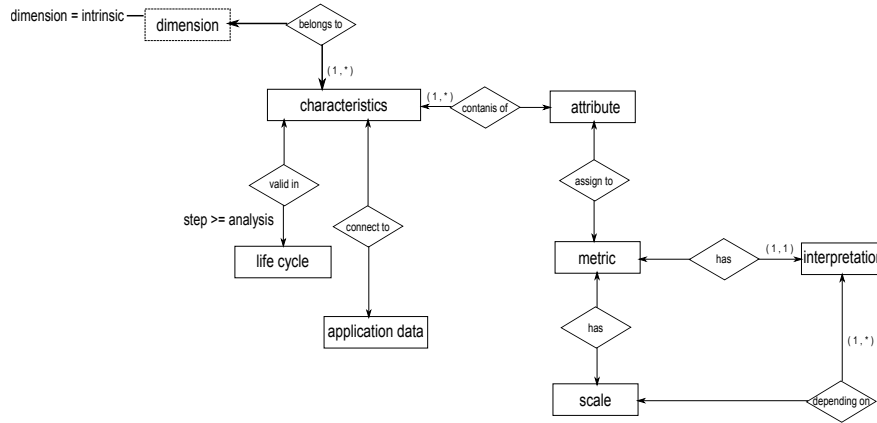


Figure 3. View schema for intrinsic quality data import

mension is restricted to intrinsic. The dashed lines around the entity type hint that there is no possibility to change the dimension for the import type. When the import type is instantiated it contains all quality metadata from the intrinsic dimension that is in the system by now. The researcher in group A can look at it and import their own quality parameter for the research data. Hence, the first part of the generic view for the import type is shown in listing 3.

```

GENERATE importIntrinsicQuality
  tcharacteristic ↦ Characteristic, tattribute ↦ Attribute, tmetric ↦ Metric,
  tscale ↦ Scale, tinterpretation ↦ Interpretation, thas_scale ↦ Has,
  tdepending_on ↦ DependingOn, tassigned_to ↦ AssignedTo,
  tconsist_of ↦ ConsistOf, tlifecycle ↦ Lifecycle, tconnect_to ↦ ConnectTo,
  tvalid_in ↦ ValidIn
FROM
  tcharacteristic = characteristics
  tattribute = attribute
  tlifecycle = lifecycle
  ...
  tvalid_in = valid in
WHERE
  dimension = 'intrinsic'
  step ≤ 'analysis'

```

Listing 3: Quality import view first part

After defining the structure, the next step is the definition of the functionality of the import type. Due to the extent of the paper at hand, only two functions will be described shortly. First, there is a function to add a new attribute to an existing quality characteristic. A new object from type  $t_{attribute}$  and an object from type  $t_{characteristic}$  are given to the function. The second function should update

the metric  $m_1$  of an attribute to a new metric  $m'_1$ . Therefore, this function will get two metrics from type  $t_{metric}$  and an existing attribute from type  $t_{attribute}$ . Thus, the *Content Type* from listing 3 can be extended by listing 4.

```

EXTEND importIntrinsicQuality
  BY FUNCTIONS manipulationFunction
    updateAttributeMetric (OldMetric , NewMetric , Attribute )
      STORED PROCEDURE := ...
  BY FUNCTIONS importFunction
    insertNewAttrToCharacteristic ( Attribute , Characteristic )
      STORED PROCEDURE := ...

```

Listing 4: Quality import view first part

With this definition, an import type for intrinsic quality parameter is created. This type can be instantiated over the database and the object can be imported into a container from a member of the research group. This example shows how easily a new content type can be created for a system. In the same way, export and harness types can be created.

## 5. Related Work

An overview of data quality management can be found in the Book of Batini and Scannapieco [1] or Lee et al. [11]. They describe different ideas, concepts, methodologies and techniques that can be used, to realise good data quality management. In the book of McGlilvray [12], a business point of view on data quality is given. He describes ten steps an enterprise can execute to get a good quality data management. The measurement of data quality in the area of data mining is given in [8]. This book summarizes different research papers for the measurement rule quality of data. A more general work for the measurement of quality parameter can be found in [17]. This book presents a large amount of different quality parameters and measures of them. Additionally, concepts for the management are given.

Moreover, a lot of different conceptual quality models can be found that define quality dimensions, characteristics and attributes. One of the important models is defined in the work of Wang and Strong [23]. The model has had a great influence on the quality data model from the DGIQ (Deutsche Gesellschaft für Informations und Datenqualität e.V.) [6], as well as on the master data management, for example, in [14]. Wang and Strong define their model by a practical survey in different industries. Within the work, four characteristics of data quality are identified and dimensions of quality are assigned to these characteristics. The four characteristics *intrinsic*, *contextual*, *representational*, *accessibility* are identified. The *intrinsic* characteristic contains all quality dimensions that are used to describe the quality of data within the whole system. It covers the dimension of

*believability, accuracy, objectivity* and *reputation*. The contextual characteristic covers all dimensions that describe how the data quality is in the context of a problem of a data set user group (DUG). *Relevancy, timeliness, completeness, appropriated amount of data* and *value-added* are the dimensions that are assigned to the contextual characteristic. The representational characteristic contains information about the format and meaning of the data. Therefore, *interpretability, ease of understanding, representational consistency* and *concise representation* are the dimensions of this characteristic. The accessibility characteristic covers information how good a DUG can access and how secure the data is within the system. Thus, *accessibility* and *access security* are the important dimensions.

Another quality model can be found in the SQuaRE (Software product Quality Requirements and Evaluation) initiative. This is a set of standards from the ISO/IEC that should replace the ISO/IEC 9126 quality standard. It consists of the ISO/IEC 250xx ISO/IEC standards like the ISO/IEC 25012 data quality model. Like the Wang and Strong model the quality is described through quality characteristics, here called categories. A category is described with characteristics. The characteristics build a hierarchy and their role can change over time depending on the life cycle of the product. Overall, there are 10 main characteristics and 27 sub-characteristics within the standard. Some of the characteristics are the same as in the Wang and Strong model, for example, reliability, and some are new, for example, maintainability of data. Different to the Wang and Strong model the attributes must have a metric that specifies the value of the parameter. A metric has a specific scale, such as numeric values or classifications, for example, acceptable and unacceptable. Another distinction to the Wang and Strong model is that the dynamic changes of the quality characteristics are explicit under consideration of the standard, for example, a quality attribute is added to a characteristic that was not known or not considered in the past. Furthermore, the standard offers three views of product quality from a DUG to a product. The *external* quality covers all characteristics that are relevant from an external view on the data, for example, the believability of data to external DUG's. *Internal* quality is defined over all quality characteristics that are relevant for an internal view on the quality. An example for this is the believability of data to an internal development team. As you can see here, the same quality characteristics can be used for different views. In contrast to the Wang and Strong model every category and it's characteristics depend on the DUG. Quality *in use* shows the view of the user to the quality of the product, e.g. the security of the data within the system. In contrast to external and internal characteristics, quality in use characteristics has no subcategories.

The importance of quality is already well-known in other areas. In [9], a framework for software quality can be found. The work is based on the SQuaRE model and presents a generic approach for the development of high quality software. For the area of *Software as a Service* (SaaS) [24] presents a Quality framework. The framework can be used to evaluate the quality of a SaaS Service. Especially the security, the *Quality of Service* (QuS) and the software quality of a Service can be evaluated with this framework. Furthermore, there are quality data frameworks

for special areas. For example, [25] presents a semantic framework for the accumulation of medical research data. The framework uses ontologies and a rule-based system to annotate quality data to aggregated patient records.

## 6. Conclusion and Future Work

This paper presents a system for managing quality data at the object level. We develop a general metadata quality component based on defining quality data as one class of metadata. This component can be associated to every other database component in the system. Components are supported by Content Types and Information Containers for exchange of data among components and retrieval of data from components. We have introduced Content Types for quality management, e.g. import types.

Our approach to quality management can be extended by adaptation features for Information Containers and envelope techniques for characterisation of metadata. The general interface to such systems is an open question. A case study for this approach is currently in preparation. We have used quality management as a showcase for general metadata management.

## References

- [1] C. Batini and M. Scannapieco. *Data Quality: Concepts, Methodologies and Techniques*. Springer-Verlag New York, Inc., 2006.
- [2] N. Carriero and D. Gelernter. Linda in Context. *Commun. ACM*, 32(4):444–458, 1989.
- [3] N. Carriero, D. Gelernter, and J. Leichter. Distributed Data Structures in Linda. In *POPL*, pages 236–242. ACM Press, 1986.
- [4] T. Feyer, K.-D. Schewe, and B. Thalheim. Conceptual Design and Development of Information Services. In *ER*, Lecture Notes in Computer Science, pages 7–20. Springer, 1998.
- [5] D. Forschungsgemeinschaft. Empfehlungen zur gesicherten aufbewahrung und bereitstellung digitaler forschungsprimärdaten., 2009.
- [6] D. G. für Informations und Datenqualität e.V.t e.V. <http://www.dgiq.de/>.
- [7] D. Gelernter. Multiple Tuple Spaces in Linda. In E. Odijk, M. Rem, and J.-C. Syre, editors, *PARLE (2)*, volume 366 of *Lecture Notes in Computer Science*, pages 20–27. Springer, 1989.
- [8] F. Guillet and H. J. Hamilton, editors. *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*. Springer, 2007.
- [9] H. Jaakkola and B. Thalheim. A framework for high-quality software design and development: a systematic approach. *IET Software*, 4(2):105–118, 2010.
- [10] K. Jannaschk, C. A. Rathje, B. Thalheim, and F. Förster. A Generic Database Schema for CIDOC-CRM Data Management. In J. Eder, M. Bieliková, and A. M. Tjoa, editors, *ADBIS (2)*, volume 789, pages 127–136, 2011.

- [11] Y. W. Lee, L. L. Pipino, J. D. Funk, and R. Y. Wang. *Journey to Data Quality*. The MIT Press, 2009.
- [12] D. McGilvray. *Executing Data Quality Projects: Ten Steps to Quality Data and Trusted Information*. Morgan Kaufmann, 2010.
- [13] R. D. Nicola, G. L. Ferrari, and R. Pugliese. KLAIM: A Kernel Language for Agents Interaction and Mobility. *IEEE Trans. Software Eng.*, 24(5):315–330, 1998.
- [14] R. Scheuch, T. Gansor, and C. Ziller. *Master Data Management: Strategie, Organisation, Architektur*. Dpunkt.Verlag GmbH, 2012.
- [15] K.-D. Schewe and B. Thalheim. Component-driven engineering of database applications. *APCCM '06 Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling*, 53:105–114, 2006.
- [16] J. W. Schmidt and H.-W. Sehring. Dockets: Model Adding Value Content. In *Proceedings of the 18th International Conference on Conceptual Modeling, volume 1728 of Lecture Notes in Computer Science*, pages 248–262, 1999. Springer-Verlag.
- [17] L. Sebastian-Coleman. *Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework*. Newnes, 2012.
- [18] L. Sebastian-Coleman. *Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework*. Morgan Kaufmann Publishers Inc., 2013.
- [19] D. Taniar and J. W. Rahayu. *Web information systems.*, chapter Structural media types in development of data-intensive web information systems, pages 34–70. IDEA Group, 2004.
- [20] B. Thalheim. Web Information Systems Analysis, Design, Development, and Implementation of Business Sites, Collaboration Sites, Edutainment (e-Learning) Sites, and Infotainment (Information) Sites.
- [21] B. Thalheim. *Entity-Relationship Modeling: Foundations of Database Technology*. Springer-Verlag, 2000.
- [22] B. Thalheim. Component Development and Construction for Database Design. *Data & Knowledge Engineering*, 54:77–95, 2005.
- [23] R. Y. Wang and D. M. Strong. Beyond accuracy: what data quality means to data consumers. *J. Manage. Inf. Syst.*, 12(4):5–33, 1996.
- [24] P. X. Wen and L. Dong. Quality Model for Evaluating SaaS Service. In *EIDWT*, pages 83–87, 2013.
- [25] L. Zhu, K. Quach, and H. Chen. A Semantic Framework for Data Quality Assurance in Medical Research. In R. Witte, C. J. O. Baker, G. Butler, and M. Dumontier, editors, *CSWS*, volume 1054, pages 54–55, 2013.

# Designing Conceptual Database Models for Innovative Evaluation of Quality

Elvira Immacolata LOCURATOLO

*ISTI Consiglio Nazionale delle Ricerche  
Via Alfieri, 1- Pisa*

*Telephone: +39 50 3152895 Fax: +39 50 3153464*

*E-mail: [elvira.locuratolo@isti.cnr.it](mailto:elvira.locuratolo@isti.cnr.it)*

**Abstract:** A theoretical approach to determine innovative evaluation of model quality is proposed. This approach is based on two different mappings to design the same conceptual database model: the former is a vertical mapping composed of bottom-up steps. It starts from the specification of a database applications supported by a formal model and achieves a *resulting model* based on semantic data models. The latter is a horizontal mapping composed of successive model extensions. It starts from a graph of conceptual classes and achieves the resulting model of the previous mapping. Formulas for the *quantitative/numerical* evaluations of the models are introduced during the vertical mapping, whereas “formulas” for the *qualitative/conceptual* evaluations of models are introduced during the horizontal mapping. The quantitative evaluations express the costs of what has been specified/proven in terms of variable and constant cardinality. The qualitative/conceptual evaluations express the saving that you get for what has been implicitly specified/proved. The *quality measure* of the resulting model is given in terms of hidden classes. These provide indication about different aspects of the model quality.

**Keywords:** Conceptual Database Models, Model Design, Model Evaluations, Quality

## Introduction

A theoretical approach to determine innovative evaluation of model quality is proposed. This approach is based on two different mappings to design the same conceptual database model: the former is a vertical mapping which starts from the specification of a database applications supported by a formal model [1], and through bottom-up steps achieves a *resulting model* based on semantic data models [3]; the latter is a horizontal mapping, which starts from a graph of conceptual classes and through successive extensions achieves the resulting model of the previous mapping. This model simplifies the conceptual model of ASSO, a database design methodology for the achievement of conflicting quality desiderata [9;11].

The vertical mapping is exploited as a means to evaluate the following aspects of quality:

- *Correct* definition of the resulting model;
- Achievement of the ASSO model desiderata;

- *Quantitative/numerical evaluation* of the resulting model in terms of state specification and consistency costs.

The horizontal mapping is exploited as a means to:

- Justify the *correct* definition of the resulting model and the achievement of the ASSO model *quality desiderata*;
- Give a *qualitative/conceptual evaluation* of the resulting model in terms of hidden classes.

The ASSO model desiderata are:

- *Easiness of use* and *Flexibility*, i.e., the resulting model is *easy to use* and *to modify*;
- *Consistency*, i.e., the resulting model is a formal model whose *consistency* can be proved.

Formulas for the *quantitative/numerical* evaluations of the models can be introduced step by step during the vertical mapping, whereas “formulas” for the *qualitative/conceptual* evaluations of models can be introduced step by step during the horizontal mapping. The quantitative evaluations express the costs of what has been specified/proven in terms of variable and constant cardinality. The qualitative/conceptual evaluations express the saving that you get for what has been implicitly specified/proved. The quantitative evaluations can be determined starting from the corresponding qualitative evaluations. The vice-versa is not possible.

The *quality measure* of the resulting model in terms of hidden classes, called *conceptual measure of quality*, provides indication about all the considered aspects of quality.

The next section is concerned with the quality evaluation of modeling methods.

## 1. Background

The techniques proposed for the evaluation of modeling methods can be classified in the following main categories.

- *Theoretical*, such as ontological evaluations, metric analysis, cognitive analysis;
- *Empirical*, such as experiments, surveys and case studies;
- *Mixed*, such as theoretical approaches with empirical evidence.

Modeling is used to compare diagrams in knowledge representation and to highlight both commonalities and differences of underlying principles in enterprise modeling, requirements modeling and design modeling [8].

Meta-modeling is a key factor on which the design of models for the achievement of quality is based. The design of these models is particularly difficult when the quality desiderata are conflicting and you do not want to discriminate one of them to the advantage of the others. In this case, innovative approaches of modeling are required to

achieve the conflicting desiderata. An example of these methods is ASSO, a methodology of conceptual database design which ensures *easiness* in specifying the conceptual schema, *flexibility* in reflecting the changes occurring in real life, *consistency* between static and dynamic modeling, *correctness* of the logical schema and *efficiency* in accessing and storing information. This methodology differs from the most common database design methodologies [2; 4; 7]; for the coexistence of all the above quality desiderata.

### 1.1. Classes in is-a relationship

The ASSO model is defined by an oriented acyclic graph of classes in *is-a relationship*. An example of these classes is presented in Figure 1. The following properties hold:

- *Classification*: each node of the graph (<person> with attribute {income}, <employee> with attribute {salary} and <student> with attribute {identifier} is a class. A node linked with a higher-level node is a class, called *specialized class*. The graph of Figure 1 includes the specialized class *employee* and the specialized class *student*.
- *Attribute Inheritance*: a specialized class (for example, the specialized class *employee*) inherits all the attributes from the higher-level classes (in our example, class *person*) and may have further attributes. Thus, the attributes of the specialized class *employee* are *income* and *salary*.

Graphs of classes defined through the is-a relationship can be supported by both semantic data models and object systems. In the following, graphs of classes supported by semantic data models are called *conceptual graphs*, whereas graphs of classes supported by object systems are called *object graphs*. Coherently with the perspective of the ODMG (Object Data Management Group), the difference between *conceptual graphs* and *object graphs* is evidenced by the following properties [6]:

- *Conceptual graph*: each object instance can belong to any class of the graph. This enhance *flexibility*. In our example, the object instances of the specialized class *employee* are a subset of the class *person* instances.
- *Object graph*: each object instance belongs to one and only one class of the graph. This enhances *efficiency*. In our example, the object instances of the specialized class *employee* and the object instances of class *person* are disjoint sets.



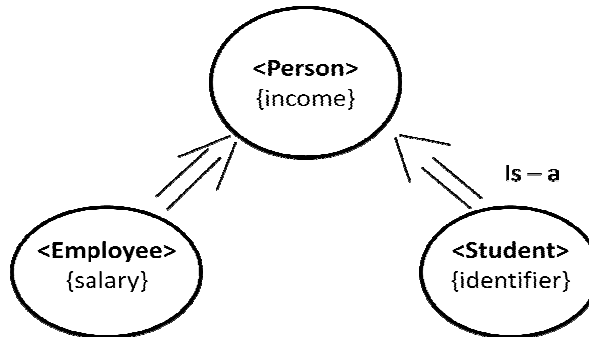


Figure 1. Graph of classes

In order to show that the object graphs limit the *flexibility* in reflecting the changes occurring in the real life, let us suppose that *student* John becomes an *employee*. In this case, the corresponding object instance must be removed from class *<Student>* and must be inserted into class *<Student•Employee>*, which is the class defined by the object intersection of class *<Student>* and class *<Employee>*. If John later on completes his studies, the corresponding object instance must be removed from the class *<Student•Employee>* and must be inserted into class *<Employee>*. On the contrary, in semantic data models, the object instance corresponding to John can be inserted into class *<Employee>* when the *Student* John becomes an *Employee*, and can be removed from class *<Student>* when John completes his studies.

To achieve the conflicting desiderata of *flexibility* and *efficiency* ASSO links conceptual graphs to objects graphs through a formal relation [12]. This relation, which is part of the ASSO refinement, is not considered in this paper. In order to result in a conceptual model achieving the conflicting desiderata of *easiness of use* and *consistency*, ASSO combines informal aspects of the database conceptual languages with formal aspects of the abstract machine model [1].

## 1.2. Abstract Machine

An *Abstract Machine* (A.M.) is defined by means of a mathematical data model and a set of operations. The data model is given by listing a set of *variables* and by writing the *invariant*, i.e., the properties of the variables. The invariant is formalized using the first-order predicate logic and a restricted version of the set-theory notation.

The Abstract Machine provides a common formal framework to model both the static and dynamic aspects of applications, i.e., the state and the operations. The operations are formalized using the *Generalized Substitution Language* (GSL), which is defined by means of two bases and some constructors. The base substitution  $x := E$  transforms the generic predicate  $R$  into the predicate obtained replacing all the free occurrences of  $x$  in  $R$  by the expression  $E$ . An example is  $[x := x+1] (x \in N) \equiv (x+1 \in N)$ . The base substitution *Skip* is the substitution that does not specify any state transformation, i.e.,  $[Skip] (x \in N) \equiv (x \in N)$ .

The constructors shown below are recursively applied to the base operations in order to define more complex operations. Specifically, the *pre-conditioned* operations

are state transformations that can be activated only when specified conditions are met; the *partial* operations are state transformations that are not defined on the whole state and the *non-deterministic* operations, are state transformations that can be implemented in different way.

### Constructors

<b>Pre <math>P</math> then <math>S</math> end</b>	Pre-conditioning
$P \implies S$	guarding
<b>choice <math>S</math> orelse <math>T</math> end</b>	bounded choice
<b>var <math>n</math> in <math>S</math> end</b>	unbounded-choice

where  $P$  is a predicate,  $S$  and  $T$  are generalized substitutions and  $n$  is a variable distinct from those of the machine state. The following axioms define the semantics of the GLS constructors in terms of weakest precondition predicate transformers [5].

$[\text{pre } P \text{ then } S \text{ end}] R$	$\Leftrightarrow$	$P \wedge [S] R$
$[P \implies S] R$	$\Leftrightarrow$	$P \Rightarrow [S] R$
$[\text{choice } S \text{ orelse } T \text{ end}] R$	$\Leftrightarrow$	$[S] R \wedge [T] R$
$[\text{var } n \text{ in } S \text{ end}] R$	$\Leftrightarrow$	$\forall n \cdot [S] R$

A special operation is the *initialization*, i.e., the operation that assigns initial values to the Abstract Machine variables. A graphical representation of the Abstract Machine is given in Figure 2, where the large oval represents the state, whereas the small ovals the operations.

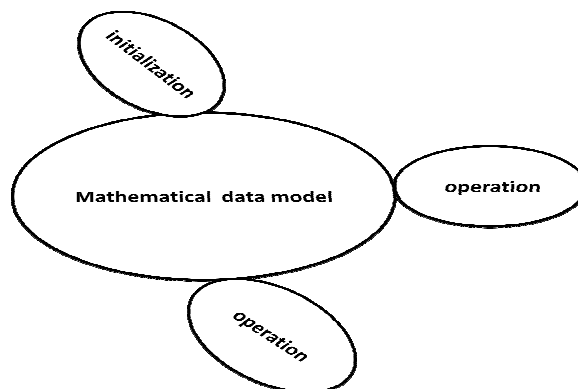


Figure 2. Abstract Machine

The axiomatic definition of the Abstract Machine permits properties of the model, such as its *consistency*, to be proved.

**Definition 1:** *A.M. consistency*

An abstract machine is consistent  $\Leftrightarrow$  the initialization establishes the invariant and each operation preserves the invariant.

Formulas *to prove* the Abstract Machine *consistency* are called *consistency obligations*. If the abstract machine has invariant  $I$  then the operation **pre  $P$  then  $S$  end** has the consistency obligation  $P \wedge I \Rightarrow [S] I$ . The *cost* of an operation correctness is related with the length of predicate  $I$ . The Abstract Machine consistency cost is defined below:

$$\text{consistency cost} = \text{initialization correctness cost} + \text{operation correctness costs}$$

In the next section, a bottom-up mapping is proposed which starts from a flat specification of a database application supported by the abstract machine model and which results in a formal conceptual graph.

## 2. Vertical Mapping

The mapping proposed in this section consists of three steps:

### 2.1. Step 1

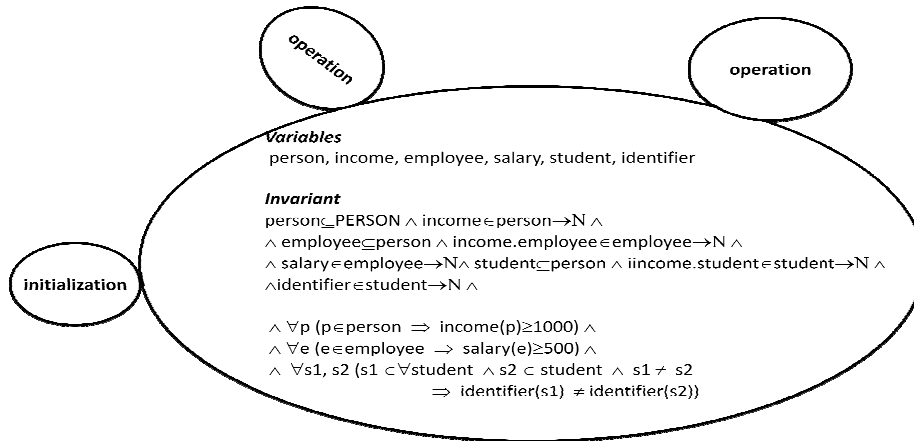
The conceptual graph represented in Figure 1 is specified exploiting a flat specification supported by the abstract machine model. This is shown in Figure 3 where the state variables denote the class names and the class attributes. The invariant consists of constraints formalizing the abstraction mechanisms of classification and *is-a relationship*. These constraints are called *implicit constraints*. In addition to the implicit constraints, application-specific constraints, called *explicit constraints*, are specified. The information specified in the state of Figure 3 is concerned with a class of *persons* and their *income*, a subclass of *working* persons and their *salary*, a subclass of *students* and their *identifier*. The *income* of each *person* is greater than or equal to 1000; the *salary* of each *employee* is greater than or equal to 500; each *student* has a unique *identifier*.

Two costs are distinguished: the *state specification cost* and the *consistency cost*. The former is defined in the following way:

$$\text{state specification cost} = \# (\text{variables and sets})$$

The *state specification cost* is *high* since it is based on the explicit specification of constraints involving the whole conceptual graph. The *consistency cost* is *high* since it is related with the lengths of the *implicit* and *explicit* constraints, which defines the A.M. invariant. In order to reduce the state specification cost, the mechanisms of

classification and is-a relationship can be declared without their explicit specifications using the database conceptual languages. The next step is concerned with this intent.



**Figure 3.** Abstract Machine specification: conceptual graph state  
 State specification cost: high  
 Consistency cost: high

## 2.2. Step 2

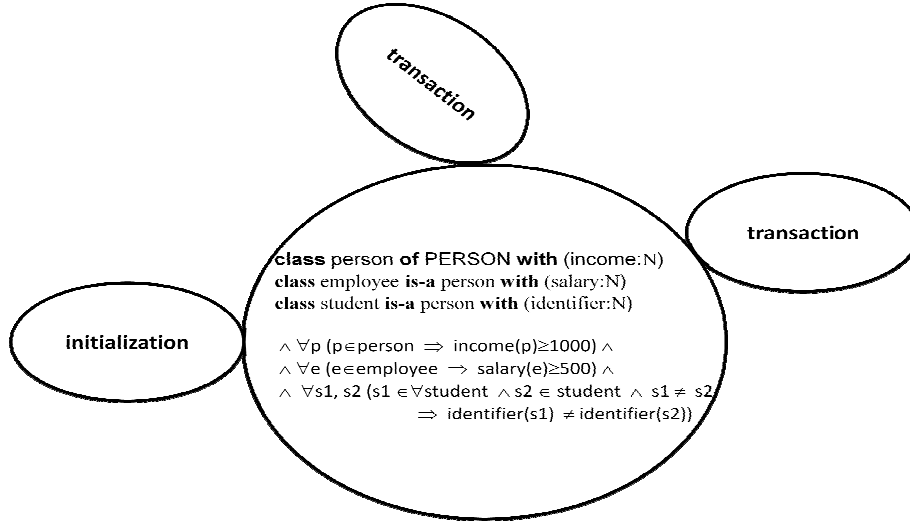
At this step, a model called *Database Schema*, which does not declare the *implicit* constraints, has been introduced. The static aspects of this model are based on the informal schemas of the database applications, whereas the dynamic aspects, i.e., the transactions, are declared as pre-conditioned, partial and non-deterministic state transformations, similar to the operations supported by the abstract machine.

### Definition 2: Database schema

*Database schema*  $\Leftrightarrow$  Abstract Machine

- whose invariant encloses constraints that, exploiting the database conceptual languages, formalize the mechanisms of *class* and *is-a relationship*.
- whose transaction specifications are not supported by any specialization mechanism.

In Figure 4, the specification of the considered conceptual graph supported by the Database Schema is given.



**Figure 4.** Database Schema specification: conceptual graph state  
 State specification cost: reduced  
 Consistency cost: high

As the mechanisms of classification and is-a relationship are implicitly specified, the *state specification cost is reduced* with respect to that of the corresponding A.M. cost. Specifically, *state specification cost = 22*.

**Definition 3:** Database Schema consistency

A database schema is *consistent*  $\Leftrightarrow$  the initialization establishes the *implicit* and *explicit* constraints and each transaction preserves the above constraints.

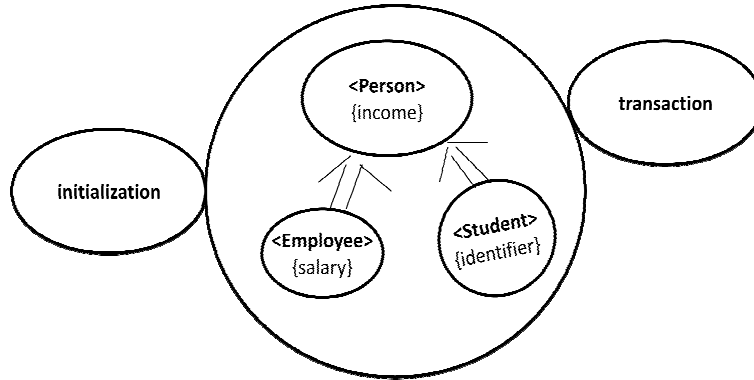
The *Database Schema* consistency cost is defined below:

$$\text{consistency cost} = \text{initialization correctness cost} + \text{transaction correctness cost}$$

As the time employed to execute an automatic proof is related with the lengths of both the *implicit* and the *explicit* constraints, and as no mechanism of specialization has been defined on the database schema, the initialization/transaction correctness proof is an *expensive process*. Similarly to the consistency cost of a generic Abstract Machine, the *consistency cost* of a *Database Schema* is *high*.

2.3. Step 3

A graphical representation of the Database Schema of Figure 4 is given in Figure 5. It evidences that the database schema state is represented as a conceptual graph, i.e., without the specification of the implicit constraints, whereas no mechanism of specialization has been introduced for the transactions. For graphical convenience, the explicit constraints have been omitted.



**Figure 5.** Database Schema specification: graphical representation

In order to reduce the *consistency costs*, the concepts of *class-machine* and *is-a\* relationship* have been introduced: the former extends the nodes of the database schema state with transactions and application constraints; the latter extends the *is-a* relationship to support the transaction specialization.

**Definition 4:** *Class-machine*

*Class-machine*  $\Leftrightarrow$  A.M. whose state specifies a class of a conceptual graph.  
In the example of Figure 5, class-machine *person*, class-machine *employee* and class-machine *student* can be defined.

**Definition 5:** *is-a\* relationship*

If class-machine *C1* is in *is-a\** relationship with class-machine *C2*, then a class-machine, called *specialized class machine C2* that inherits attributes and transactions, is defined. An *inherited* transaction of the specialized class-machine *C2* is defined through the parallel composition of a transaction on class-machine *C1* with the corresponding transaction on class-machine *C2*. This latter transaction is called *specialization* on *C2*. The initialization and the transactions on *C2* that insert objects are explicitly specialized to preserve the *is-a* constraints, whereas the remaining transactions can be implicitly specialized (See Figure 7).

The semantics for the inherited transaction *Tr* of the specialized class-machine *SC* is the following:

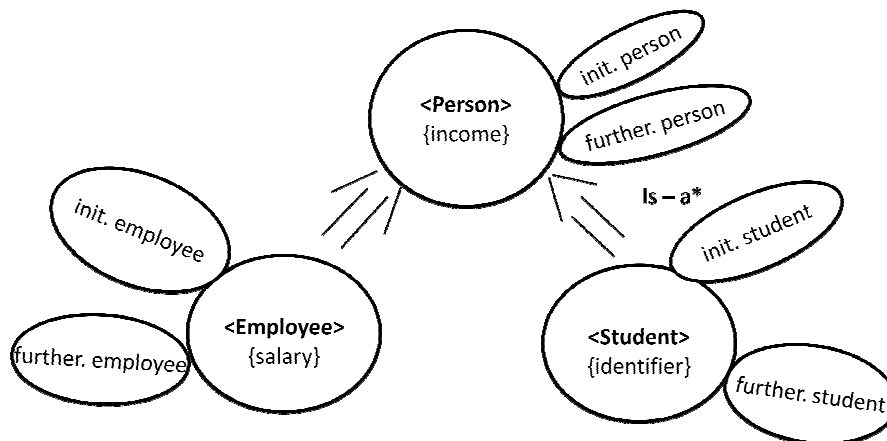
$$[Tr\ C2\ is-a^*\ C1\ (par\_list)]\ R \Leftrightarrow [Tr\ C1\ (par\_list)]\ R1 \wedge [Tr\ C2\ (par\_list)]\ R2 \wedge (is-a\ constraints \Rightarrow is-a\ constraints') \wedge R3'$$

where  $R$  is a predicate on the variables of the specialized class-machine,  $R_1$  is a predicate on the  $C_1$  variables,  $R_2$  a predicate on the  $C_2$  variables,  $R_3 = R - (R_1 \wedge R_2)$ , and  $(is-a\ constraints \Rightarrow is-a\ constraints')$  is the predicate which preserves the *is-a* constraints, i.e., the predicate which formalizes the object inclusion and attribute inheritance properties of the *is-a* relationship.

**Definition 6: Resulting Model**

The Resulting model is an oriented acyclic graph of *class-machines* in *is-a\** relationships.

Figure 6 presents the resulting model of our example. With respect to the Database Schema, the transactions are partitioned on the nodes of the resulting model. For graphical convenience, the *explicit* constraints have been omitted. As an example of transaction, a working-student not belonging to the conceptual graph is added to the three classes.



**Figure 6.** Resulting Model:example  
Consistency cost= Sum of class-machine consistency costs

The following notation is exploited to declare the root class-machine:

**Class-Machine** name of set **with** (attr-list; explicit; trans-list)

where *name* is a variable denoting the class-machine name, *set* a given set, *attr-list* the list of the class-machine attributes, *explicit* the set of application-specific constraints, and *trans-list* the list of transactions including the initialization.

```

Class-Machine person of PERSON with (income : N;
  person  $\forall p (p \in \text{person} \Rightarrow \text{income}(p) \geq 1000)$ 
  init.person () = person, income : = J, 1000 ;
  further.person (pers, inc) =
    PRE
      Pers  $\in$  PERSON-person  $\wedge$  inc  $\geq$  1000
    THEN
      ADD person (pers, inc)
    END

```

In class-machine *person*, the attributes list is defined only by *income*, and the *explicit* constraints are defined only by  $\forall p (p \in \text{person} \Rightarrow \text{income}(p) \geq 1000)$ . The class-machine *person* is not empty. Moreover, the *person* J with *income* 1000 defines the initial state. The parametric transaction inserts a person, not already belonging to the class-machine, with his/her income. In order to insert this person also in class-machine *employee* and in class-machine *student*, the following notation is exploited:

**Class-Machine** name2 is-a\* name1 **with** (attr-list; explicit; trans-list).

```

Class-Machine employee is-a* person with (salary: N;

  employee  $\forall e (e \in \text{employee} \Rightarrow \text{salary}(e) \geq 500)$ 
  employee () = employee, salary : = J, 1000
  further.employee (pers, sal) =
    PRE
      Sal  $\geq$  500
    THEN
      ADD employee (pers, sal)
    END)

class student is-a* person with (identifier: N;
  student  $\forall s_1, s_2 (s_1 \in \text{student} \wedge s_2 \in \text{student} \wedge s_1 \neq s_2$ 
     $\Rightarrow \text{identifier}(s_1) \neq \text{identifier}(s_2))$ ;
  init.student () = student, identifier: =  $\emptyset, \emptyset$ ;
  further.student (pers) =
    ANY m WHERE m  $\in$  N  $\wedge$  m  $\notin$  ran (identifier)
    THEN
      ADD student (pers, m)
    END

```

In order to justify that the consistency cost of the resulting model is improved with respect to the Database Schema corresponding cost, let us consider the example of transaction inheritance in Figure 7. The two transactions consist of two specifications:  $\text{init.person}_e \parallel \text{init.employee}_e$  and  $\text{further.person}_e \parallel \text{further.employee}_e$ . However, the  $\text{init.person}_e$  and the  $\text{further.person}_e$  need not to be proven, since already proved for the root class-machine of the graph.



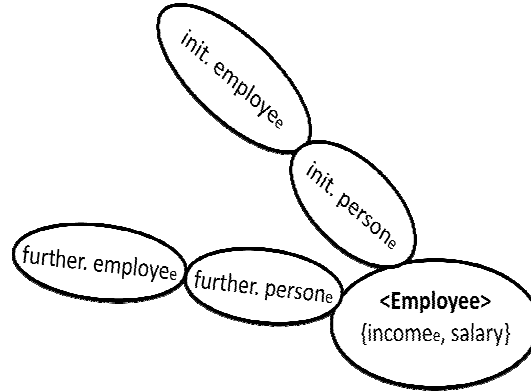


Figure 7. Transaction inheritance

**Property 3:** *consistency cost*

The consistency cost of the resulting model is obtained by adding the consistency costs of the class-machines.

In our example:

consistency cost = class-machine *person* consistency cost + class-machine *employee* consistency cost + class-machine *student* consistency cost.

As the time employed to execute an automatic proof is related with the lengths of both the *explicit* and *implicit* predicates, the following formulas can be exploited to evaluate the *correctness cost* of a transaction:

$$\text{Cost (Skip)} = 0$$

$$\text{Cost (Basic Tr C (par\_list))} = \# (\text{explicit} \wedge \text{implicit})$$

where BasicTr C (*par\_list*) is a basic transaction not coinciding with Skip and  $\# (\text{explicit} \wedge \text{implicit})$  is the cardinality of both constants and variables of the *implicit* and *explicit* constraints of class-machine C.

The *correctness cost* of a transaction Tr is recursively defined as follows [11].

$$\begin{aligned} \text{cost ([PRE P THEN Tr C (par\_list) END] explicit)} &= \text{cost ([Tr C (par\_list)]) explicit)} \\ \text{cost ([P \Rightarrow Tr C (par\_list)] explicit)} &= \text{cost ([Tr C (par\_list)] explicit)} \\ \text{cost ([CHOICE Tr C (par\_list)] ORELSE Tr C *(par\_list) END] explicit)} &= \\ &= \text{cost ([Tr C (par\_list)] explicit)} + \text{cost ([Tr C *(par\_list)] explicit)} \\ \text{cost ([ANY y WHERE P THEN Tr C (par\_list) END] explicit)} &= \\ &= \text{cost ([Tr C (par\_list)] explicit)} \end{aligned}$$

The correctness of the further.emp (*pers, sal*) transaction specified in class **Class-Machine** employee is guaranteed by proving that the following formula is true.

$$\begin{array}{l}
\textit{implicit} \wedge \textit{explicit} \wedge \textit{pre-conditions} \Rightarrow [\textit{further.emp}(\textit{pers}, \textit{sal})] \textit{explicit} \\
\\
\Leftrightarrow \\
\textit{implicit} \wedge \textit{explicit} \wedge \textit{pre-conditions} \Rightarrow \\
\textit{sal} \geq 500 \wedge \quad \quad \quad \bigg| \quad \textit{pre-conditions} \\
\wedge \textit{employee} \subseteq \textit{PERSON} \wedge \\
\wedge \textit{salary} \in \textit{employee} \rightarrow \mathbb{N} \quad \bigg| \quad \textit{implicit} \\
\Rightarrow \\
(\textit{employee} \cup \{\textit{pers}\}) \subseteq \textit{PERSON} \wedge \\
\wedge (\textit{salary} \langle + (\textit{pers}, \textit{sal}) \rangle \in (\textit{employee} \cup \{\textit{pers}\}) \rightarrow \mathbb{N}) \wedge \quad \bigg| \quad \textit{implicit}' \\
\wedge \forall e (e \in (\textit{employee} \cup \{\textit{pers}\}) \Rightarrow (\textit{salary} \langle + \{\textit{pers}, \textit{sal}\} \rangle (e) \geq 500)) \quad \bigg| \quad \textit{explicit}'
\end{array}$$

As the *consistency cost* has been defined by adding the initialization correctness cost and the transaction correctness costs, the given formulas suffice to prove the *consistency cost* of any class-machine.

#### 2.4. Quality Achievement

The following aspects of quality have been achieved by the resulting model: *correct* model definition, *quality desiderata*, *formulas* for the numerical evaluation of reduced specification and consistency costs.

As to the *correct definition* of the model, let us observe that the resulting model has been obtained applying model transformations preserving equivalence to the initial specification (Locuratolo, 2005). The resulting model is seen as a high abstraction level where some details are implicitly specified, whereas the abstract machine is seen as a lower abstraction level where all the details are explicitly stated. The translation from the resulting model to Abstract Machines allows using B support tools. When the designer wishes to check the consistency of her/his specification, she/he invokes this automatic translation and can use the B proof obligation generator and prover [1].

As for the achievement of the *quality desiderata*, the resulting model exhibits the *flexibility* of the conceptual graphs. Further, it provides the advantages of both informal and formal notations. Similarly to the database conceptual languages, the structured database schema is an *easy to be used* model because many details explicitly specified with the formal notations are avoided. However, unlike the informal notations, the conceptual schema *consistency* can be proved.

The introduced formulas suffices to prove that the *specification cost* and the *consistency cost* of the resulting model are reduced with respect to corresponding costs of Database Schema specification and consistency.

The next section describes the horizontal mapping.

### 3. Horizontal Mapping

The mapping presented in this section is composed by three steps:

### 3.1. Step 1

The conceptual graph of Figure 8 is correctly transformed into the root class *person*, the specialized class *employee* and the specialized class *student*. The specialized class *employee* is defined by the *employee* objects, by the inherited attribute *income* restricted to the set of the employee objects, i.e., *income<sub>e</sub>*, and by the specific attribute *salary*. Analogously, the specialized class *student* is defined by the *student* objects, by the inherited attribute *income* restricted to the set of the student objects, i.e., *income<sub>s</sub>*, and by the specific attribute *identifier*. The three classes, represented in Figure 9 without any link, are independent classes that explicitly specify all and only the information enclosed in the conceptual graph of Figure 8.

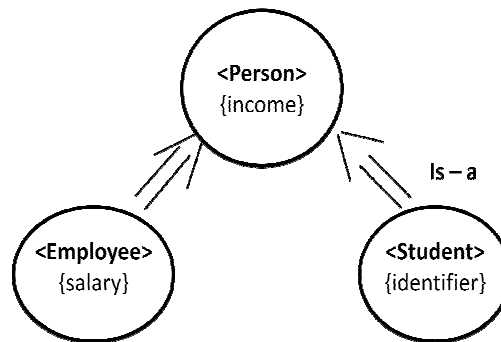


Figure 8. Conceptual graph

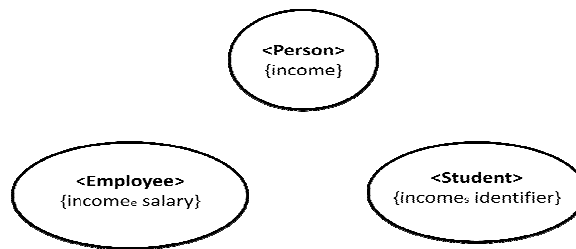


Figure 9. Root class and specialized classes

The *quality measure* of the conceptual graph in Figure 8, called conceptual measure of quality is defined by the following two classes of Figure 10:

**<Employee>** {income<sub>e</sub>}; **<Student>** {income<sub>s</sub>}

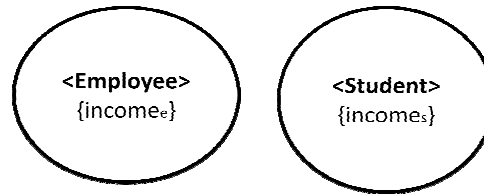


Figure 10. Quality measure

These classes, which have been obtained subtracting from the specialized classes the corresponding classes of the conceptual graph, represent the information implicitly specified through the is-a relationships.

$$\begin{aligned} \langle \mathbf{Employee} \rangle \{ \text{income}_e \} &= \langle \mathbf{Employee} \rangle \{ \text{income}_e, \text{salary} \} - \langle \mathbf{Employee} \rangle \{ \text{salary} \} \\ \langle \mathbf{Student} \rangle \{ \text{income}_s \} &= \langle \mathbf{Student} \rangle \{ \text{income}_s, \text{identifier} \} - \langle \mathbf{Student} \rangle \{ \text{identifier} \} \end{aligned}$$

The *conceptual measure of quality* expresses the saving that you get for what has been implicitly specified. More hidden classes define this measure, more semantic richness they have, the higher is the quality of the model.

### 3.2. Step 2

The conceptual graph of Figure 8 is extended with basic transactions as follows: when a student-employee is added to the conceptual graph, the basic transactions *new.person*, *new.employee* and *new.student* are defined on the corresponding conceptual graph classes. The extended conceptual graph is graphically represented in Figure 11. Similarly to the attributes, the basic transactions are inherited through the extended is-a relationship. An extended *root* class and the *extended specialized classes* of Figure 12 are generated. The inherited transaction on the extended specialized *employee/student* class is defined by the *new.person* transaction restricted to the *employee/student* objects, i.e., the *new.person|new.persons*, composed through the  $\parallel$  operator with the *new.employee/new.student* specialization. The specialized classes are independent classes that explicitly specify all and only the information enclosed in the extended conceptual graph.

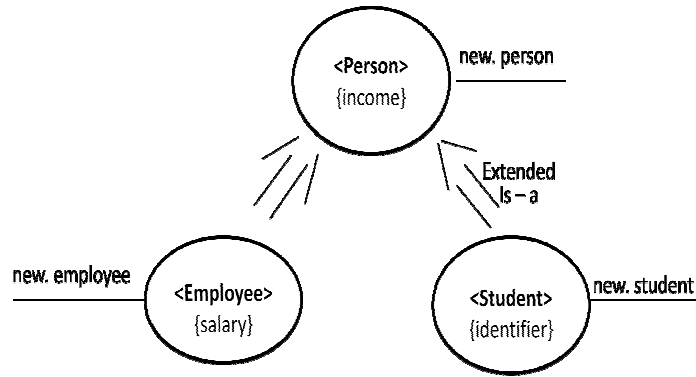


Figure 11. Extended conceptual graph

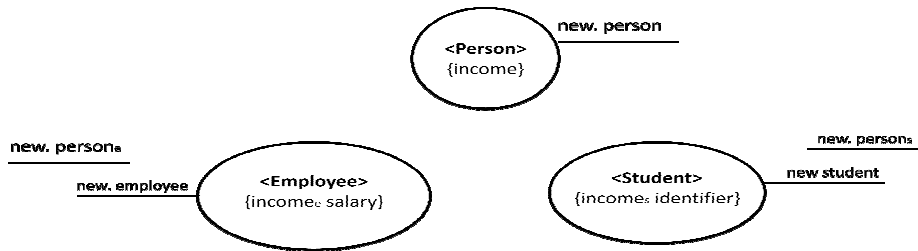


Figure 12. Extended root class and extended specialized classes

The *quality measure* of the extended conceptual graph is defined by the two extended classes in Figure 13:

$$\langle \text{Employee} \rangle \{ \text{income}_e, \text{new.person}_e \}; \langle \text{Student} \rangle \{ \text{income}_s, \text{new.person}_s \}$$



Figure 13. Quality measure of the extended conceptual graph

These classes, which have been obtained subtracting from the extended specialized classes the corresponding extended classes, represent the information implicitly specified through the extended is-a relationships.

$$\langle \text{Employee} \rangle \{ \text{income}_e, \text{new.person}_e \} = \langle \text{Employee} \rangle \{ \text{income}_e, \text{salary}, \text{new.person}_e \parallel \text{new.employee} \} - \langle \text{Employee} \rangle \{ \text{salary}, \text{new.employee} \}$$

$\langle \text{Student} \rangle \{ \text{income}, \text{new.persons} \} = \langle \text{Student} \rangle \{ \text{income}, \text{salary}, \text{new.persons} \parallel \text{new.employees} \} - \langle \text{Student} \rangle \{ \text{identifier}, \text{new.students} \}$

**Property 4:** *quality measure*

The quality measure of the extended conceptual graph is the extension of the quality measure of the conceptual graph.

3.3. *Step 3*

The extended conceptual graph is further extended with general transactions enclosing an *initialization* and application constraints. The transactions are defined by applying the GSL constructors to the basic transactions. The initialization assigns the initial values to the variables of the model. The extended is-a relationship, further extended to support the transaction/initialization specialization, is called *is-a\** relationship. Similarly to attributes, the transactions are inherited. The further extended conceptual graph represented in Figure 14 is correctly transformed in the further extended *root* class and the *further extended specialized classes* of Figure 15. These classes are independent classes, which explicitly specify all and only the information enclosed in the further extended conceptual graph of Figure 14.

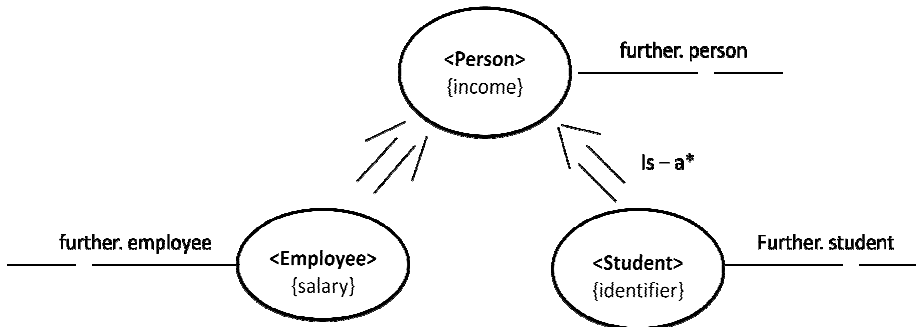


Figure 14. Further extended conceptual graph

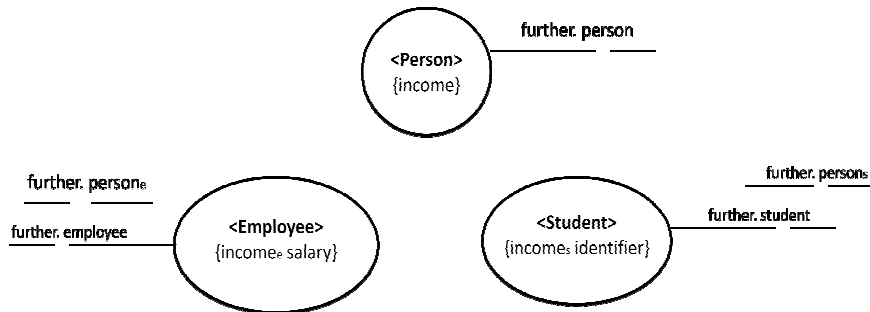
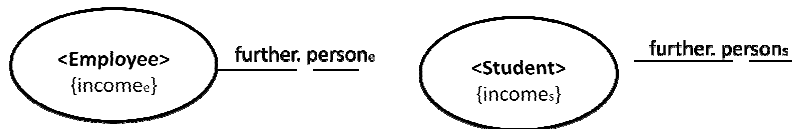


Figure 15. Further Extended root class and further extended specialized classes

The *quality measure* of the further extended conceptual graph in Figure 14 is defined by the following two further extended classes of Figure 16. These represent the information implicitly specified through the further extended is-a relationships:

**<Employee>** {income<sub>e</sub>, further.person<sub>e</sub>}; **<Student>** {income<sub>s</sub>, further.person<sub>s</sub>}



**Figure 16.** Quality measure of the Further extended conceptual graph

These classes are obtained subtracting from the further specialized classes the corresponding classes of the further extended conceptual graph, i.e.,

**<Employee>** {income<sub>e</sub>, further.person<sub>e</sub>} = **<Employee>** {income<sub>e</sub>, salary, further.person<sub>e</sub> || new.employee} - **<Employee>** {salary, further.employee}

**<Student>** {income<sub>s</sub>, further.person<sub>s</sub>} = **<Student>** { income<sub>s</sub>, salary, further.person<sub>s</sub> || further.employee } - **<Student>** {identifier, further.students}

#### **Property 5:** *quality measure*

The quality measure of a further extended conceptual graph is the extension of the quality measure of an extended conceptual graph.

The approach can be generalized to general conceptual graphs/extended conceptual graphs/further extended conceptual graphs.

## **4. Discussion**

The approach proposed to innovative evaluation of model quality is based on two different mappings to design the same conceptual database model: the former starts from the abstract machine model and through two vertical steps restrict the set of all the possible abstract machines to those formalizing conceptual graphs extended with specialized transactions. The emphasis of this approach is on the *Abstract Machine* model. The following aspects of quality have been achieved: *correct* model definition, *quality desiderata*, *formulas* for the numerical evaluation of reduced costs of specification, and consistency of the resulting model.

The latter mapping starts from a conceptual graph that can be used in information systems, software engineering and knowledge engineering, and design a graph of class-machines through two horizontal steps. The emphasis of this approach is on the *Conceptual Graphs*. The resulting model of the horizontal mapping coincides with the resulting model of the vertical mapping. Thus the correct definition of the model and the achievement of the quality desiderata are guaranteed not only for the vertical mapping but also for the horizontal mapping.

The conceptual measure of quality can be determined not only for the resulting model, but also for the initial conceptual graph, for the extended conceptual graph, and

in general for the resulting models. This *measure* which is defined by a set of hidden classes/extended classes/class-machines, which are implicitly specified within conceptual graphs/extended conceptual graphs/further extended conceptual graphs expresses the saving that you get for what has been implicitly specified/proved. The amount of saving increases when the number of classes and the complexity of specification increase.

To discuss the relationships between the two evaluations of the resulting model, let us observe that, starting from the conceptual measure of quality, numerical costs of class specifications and consistency proofs can be determined. These express the savings that you get by specifying the resulting model with class-machines rather than with the Database Schema. Vice-versa, a *conceptual measure of quality* cannot be determined starting from numerical costs of specifications and consistency proofs.

The implementation of tools for the determination of conceptual measures of quality exploit the property that the quality of an extended conceptual graph is the extension of a previous determined conceptual measure.

## 5. Conclusions and further developments

Two different mapping to design the same conceptual model, called *resulting model*, are proposed. The *resulting model*, which integrates aspects of semantic data models with aspects of the B formal method simplifies the conceptual model of the ASSO methodology. The former mapping starts from the abstract machine model, and through vertical steps reaches first the Database Schema model, and then the resulting model. The latter mapping starts from a conceptual graph that can be used in information systems, software engineering and knowledge engineering, and through two horizontal steps reaches the previous resulting model. Quantitative and qualitative evaluations of this model have been provided. Further developments of this research are concerned with the optimization of the ASSO methodology.

## References

- [1] Abrial, J. R. (1989). A Formal Approach to Large Software Construction. In: *Mathematics of Program Construction. Lecture Notes in Computer Science*, 375 (pp 141-158). Berlin: Springer Verlag.
- [2] Batini, C., Ceri, S., & Navathe, S. B. (1992). *Conceptual Database Design: An Entity-Relationship Approach*. Redwood City, California: Benjamin Cummings.
- [3] Cardenas, A. F., & McLeod, D. (1990). *Research Foundations in Object-Oriented and Semantic Database Systems*. Englewood Cliffs, NJ 07632: Prentice Hall.
- [4] Ceri, S., & Fraternali, P. (1997). *Database Applications with Objects and Rules*. Edinburgh Gate Harlow Essex, UK: Addison Wesley Longman.
- [5] Dijkstra, E.W. & Scholten, S. (1990). *Predicate Calculus and Program Semantics*. New York: Springer-Verlag
- [6] Elmasri, R., & Navathe, S.B. (2003). *Fundamentals of Database Systems*. Addison Wesley.
- [7] Jarke, M., Mylopoulos, J., Schmidt, W. & Vassiliou, Y. (1992). DAIDA: An Environment for evolving Information Systems. *ACM Trans on Information*



- Systems*, 10(1), 1-50.
- [8] Krogstie J., Halpin T. & Keng S. (Eds). (2005). *Information Modeling Methods and Methodologies*. Hershey PA: Idea Group Publishing.
  - [9] Locuratolo, E. & Matthews, B. (1999). On the relationship between ASSO and B. In: H. Jaakkola, H. Kangassalo & E. Kawaguchi (Eds), *Information Modelling and Knowledge Bases X*. (pp.235-253). IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington.
  - [10] Locuratolo, E. (2009). Database Design Based on B. In: J. Erickson (Ed.), *Database Technologies: Concepts, Methodologies, Tools, and Applications*, (4 Vols) (pp. 400-456). Hershey PA: IGI Global.
  - [11] Locuratolo, E. (2011). Meta-Modeling to design the Structure Database Schema. In: K. Siau, R. H. L. Chiang, & B. C. Hardgrave (Eds.): *Systems Analysis and Design: People, Processes, and Projects* (pp. 195–215). (Advances in Management Information Systems, vol. 18). Armonk N.Y. : M. E. Sharpe.
  - [12] Locuratolo, E. (2013). A Constructive Approach for Conceptual Database Design. In: K.A. Buragga & N. Zana, (Eds), *Software Development Techniques for Constructive Information Systems Design*. Hershey PA: IGI global.

# Relating Concept Theory to Computer Science

Elvira LOCURATOLO<sup>1</sup> and Jari PALOMÄKI<sup>2</sup>

<sup>1</sup>ISTI CNR, Via G. Moruzzi, 1/56124 Pisa, Italy

Tel. +39 050 315 2895, Fax +39 050 315 2810

e-mail: [elvira.locuratolo@isti.cnr.it](mailto:elvira.locuratolo@isti.cnr.it)

<http://www.isti.cnr.it/People/E.Locuratolo>

<sup>2</sup>Tampere University of Technology/Pori, Department of Information Technology,  
Pohjoisranta, P.O Box 300, FI-28101, Finland

Tel. +358 40 8360025, Fax +358 9 4528318

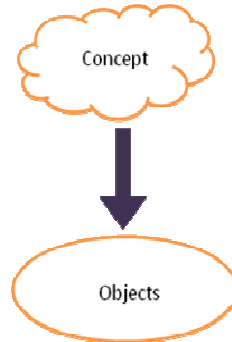
e-mail: [jari.palomaki@tut.fi](mailto:jari.palomaki@tut.fi)

**Abstract.** An algorithm is introduced in concept theory to design concept structures related to object classes/categories supported by computer systems. Although concept theory has a formal background, these algorithms are not yet available. The approach is supported by a methodology which starts from algorithms of object decomposition proposed in computer science and reaches an algorithm of concept construction related to class/categories of objects.

**Keywords:** Concept, intension, extension, partitioning, conceptual modeling, database models, mapping algorithms, preservation

## Introduction

In Concept Theory [4,14] we distinguish between an *intension* and an *extension* of a concept. Intension refers to the information content of the concept, whereas extension refers to the set of objects which fall under the intension. The unidirectional link of Figure 1 shows that each concept has as its extension one and only one class/category of objects. On the other hand, a class/category of objects can be the extension of many different concepts, for example, a set of apples can be extension of the concept “apple”, of the concept “fruit”, and so on. *Concepts* are related to each other by means of an *intensional containment relation*. Using this relation, it is possible to define concept operators and concept structures. *Objects* are organized into *classes/categories*. Concepts exist independently from classes/categories of objects, thus the following two levels of representation must be taken into consideration: an *intensional concept* level and an *extensional set-theoretical* level. Although concept theory has a formal background, algorithms to design concept structures related to classes/categories of objects supported by computer systems are not yet available.



**Figure 1.** From concept to objects

In conceptual database modeling and software engineering applications, the concept level and the set-theoretical level are collapsed into a single level. The formal aspects of these applications are treated using the set-theory or its extensions. The link between concept theory and computer science is thus at the set-theoretical level. Algorithms of mappings from classes of objects supported by semantic data models, called *semantic classes*, to classes of objects supported by object systems, called *object classes* have been proposed in computer science. Both these models are defined as acyclic oriented graphs of classes. In semantic classes, each object instance can belong to many different classes, thus enhancing *flexibility*. In object classes, each object instance can belong to one and only one class, thus enhancing *efficiency*. By *partitioning*, we mean a class of algorithms that maps graphs of semantic classes to graphs of object classes. These algorithms are based on the decomposition of objects and on the inheritance of attributes. The first of these algorithms, called the *partitioning method* [7], was designed to combine the conflicting quality desiderata of *flexibility* into modifying a database schema supported by semantic classes, and the *efficiency* of object data systems. As a consequence of this transformation, database applications can be specified with flexibility by referring to the conceptual schema, while the obtained implementations can exploit the efficiency provided by object database systems.

The partitioning algorithms cannot be applied at the concept level since a unidirectional link holds from concepts to objects, however, an algorithm to define concept structures related to class of objects can be designed. To reach this goal, intensional operators having corresponding set theory partitioning operators are firstly introduced; an initial concept structure is then identified and an algorithm of concept construction working on the initial concept structure is finally proposed. As a result of this approach, a concept network is obtained. The leaves of this network can be correctly mapped to graphs of classes supported by object data systems. The approach is *complete* with respect to both: *concepts and classes*. *Concept completeness* is distinguished from *class completeness*. Exploiting this approach a network of concepts, called the *ontology for database preservation*, has been constructed and mapped to the Universe of Discourse and to the database models [8].

This paper focuses on the methodology exploited to design the algorithm. It is organized as follows: Section 1 provides background information on the partitioning algorithms and on the concept theory; a partition algorithm is provided. Section 2 provides the methodology exploited to design the algorithm of concept construction.

Section 3 provides a discussion. Conclusions and further developments are included in Section 4.

## 1. Background

This section provides the background of the paper. Section 1.1 concerns mappings in database design. Section 1.2 includes the partitioning algorithm of maximum steps and Section 1.3 is about the items connected to a concept and to the concept theory.

### 1.1. Mappings in database design

In database design, *mappings* are required to translate conceptual schemas into schemas processed by some database management systems. *Mapping* conceptual data models, such as ER models, to logical data models has been widely investigated [3]. Semantic data models are appropriate models for conceptual design since they allow the representation of database objects close to the real world objects, and the ability to reflect the changes occurring in real life with flexibility [2]. However, as complete database management tools, semantic data models have never been implemented efficiently [14]. *Transformations* between conceptual models, above all entity relationship diagrams and UML, into a data model for a relational, object relational, object oriented and XML databases are presented in [1]. Objects models, which have abstraction mechanisms similar to those of semantic data models, have reached a remarkable level of efficiency [2]. A *formal mapping*, called the *Partitioning Method*, has been proposed in [9] to achieve the flexibility of semantic data models and the efficiency of object systems, two conflicting quality desiderata. An acyclic oriented graph of classes supported by semantic data models has been mapped to classes that can be supported by object systems. This approach is based on recursive graph decompositions until all and only disjoint classes are obtained. In line with the Object Data Management Group, ODMG, the best result in the engineering of object database systems can be explained by observing that in semantic data models, each object instance can belong to any class of graph thus enhancing flexibility. In object systems, on the other hand, each object instance belongs to one and only one class, thus enhancing efficiency while limiting flexibility. An algorithm which permits the database designer to choose from a database schema supported by a multi storage object model, a database schema supported by a single storage object model, or a database schema supported by a relational model is provided in [7].

The Partitioning method was designed by introducing difficult operators of graph decompositions. The final decomposition results in all disjoint classes which are not recomposed into a graph of object classes. In order to overcome these drawbacks, new algorithms of partitioning were introduced: these algorithms exploit the expressive power of the labels to identify the object classes. Further, the disjoint classes are recomposed into graphs supported by object systems. It is important to show that a class of these partitioning algorithms exists [16,10]. All of them are based on the decomposition of objects and on the inheritance of attributes. These algorithms provide the same sets of disjoint classes as an output. However, the intermediate steps are defined by different classes. In what follows, an algorithm of partitioning, called *the algorithm of maximum steps*, is provided. This algorithm generates the maximum number of intermediate classes before reaching the classes supported by the object systems.

### 1.2. The algorithm of maximum steps

The partitioning algorithms are applied to graphs of semantic classes. As an example, let us first refer to the graph of semantic classes shown in Figure 2.

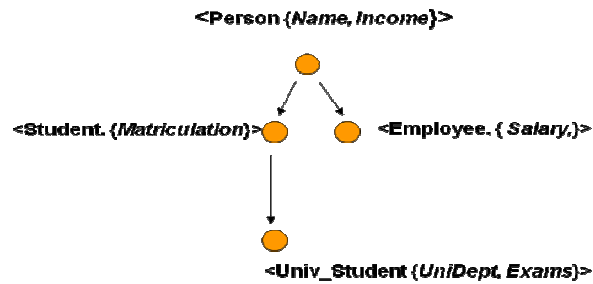


Figure 2. Semantic Classes

In this graph, the nodes are classes. A class is composed of a set of objects and a set of attributes associated with the objects. A class has a *label*, which denotes both the class name and the set of class objects, and has a list of attributes. In our example, <Person, {Name, Income}> is the root node. The class name is Person; the list of attributes is {Name, Income}. <Student, {Matriculation}> is a class in *is-a* relationship with class <Person, {Name, Income}>. This means that the Student objects are enclosed within the Person objects and that the Student class inherits the attributes from the Person class. In our example, the Student class also has a specific attribute. The *is-a* relationship is represented as a link directed from class <Person, {Name, Income}> to class <Student, {matriculation}>. Class <Employee, {Income}> is in *is-a* relationship with class <Person, {Name, Income}>. Class <Univ\_Student, {Uni\_dept, Exams}> is in *is-a* relationship with class <Student, {matriculation}>. Hereafter, for brevity, the label <P> will denote the class <Person, {Name, Income}> the label <S> will denote the class <Student, {Matriculation}>, and so on. Classes of objects can also be linked through *is-a<sub>o</sub>* relationships.

The following differences hold among classes of objects in *is-a* / *is-a<sub>o</sub>* relationships:

- A class <Y> is in *is-a* relationship with class <X>, if the objects of class <Y> are enclosed within the objects of class <X>. Class <Y> inherits the attributes from class <X> and can also have specific attributes.
- A class <Y> is in *is-a<sub>o</sub>* relationship with class <X>, if the objects of class <Y> are disjoint from the objects of class <X>. Class <Y> inherits the attributes from class <X> and can also have specific attributes.

Class <S> is in *is-a* relationship with class <P>. Class <P-S> = <Person-Student, {Name, Income}> is defined by the set of persons that are not students. The attributes are only those of class <P>. Class <P-S> and class <S> are disjoint classes. Class <P-S> is in *is-a<sub>o</sub>* relationship with class <S>. Class <S> inherits the attributes from class

<P-S> and has a specific attribute. The *is-a<sub>o</sub>* relationships are oriented in opposite directions compared to the *is-a* relationships.

A partitioning algorithm, called the *algorithm of maximum steps*, that generates the maximum number of intermediate classes before reaching the classes supported by the object systems, is the following:

```

Algorithm of maximum steps ( $A_S$ )
Begin
If  $A_S = \langle \text{root} \rangle$  then
  Return  $A_o = A_s$ 
Else
If  $A_S = \langle \text{son} \rangle \text{ is-a } \langle \text{root} \rangle$  then
  Return
   $A_o = \langle \text{root-son} \rangle \text{ is-a } \langle \text{root} \rangle$ 
Else begin
  Decompose  $A_S$  in  $A_{S1}$  and  $A_{S2}$ 
   $A_{o1} = \text{Algorithm of maximum steps}$  ( $A_{S1}$ )
   $A_{o2} = \text{Algorithm of maximum steps}$  ( $A_{S2}$ )
   $A_o = \text{Merge}$  ( $A_{o1}, A_{o2}$ )
Return  $A_o$ 
End

```

The algorithm of maximum steps is based on the following points:

- Direct solution of the problem applied to two elementary cases.
- Decomposition of the problem in two independent sub problems of the same type.  $A_S$  is the graph of semantic classes, also called *specialization hierarchy of semantic classes*. This specialization hierarchy is decomposed into the two specialization hierarchies  $A_{S1}$  and  $A_{S2}$ . The roots of  $A_{S1}$  and  $A_{S2}$  define a partition of the  $A_S$  root. The partition is obtained by set difference and set intersection between the  $A_S$  root and the label of the most-left child.
- Recursive solution of each sub-problem.
- Composition of the two sub-problem solutions to obtain the global solution. This is obtained by linking the roots of the two sub-problem solutions through the *is-a<sub>o</sub>* links.
- Implicit information is specified through the root labels: only the attributes of class <X> are associated with a node labeled by <X-Y>, whereas the attributes of all the classes <X>...<Y> are associated with a node labeled by <X  $\cap$  ...  $\cap$  Y>.

An example of applicability of this algorithm is provided using Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, Figure 8 and Figure 9.

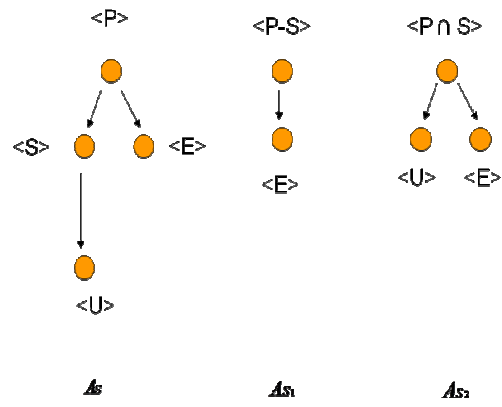


Figure 3. Decomposition of semantic classes: first step

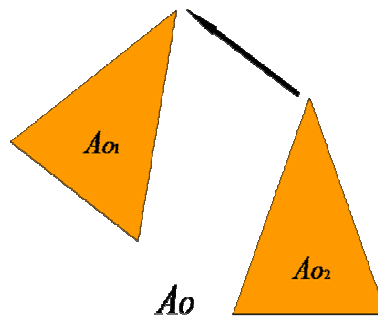


Figure 4. Merge Procedure:  $A_o$  definition

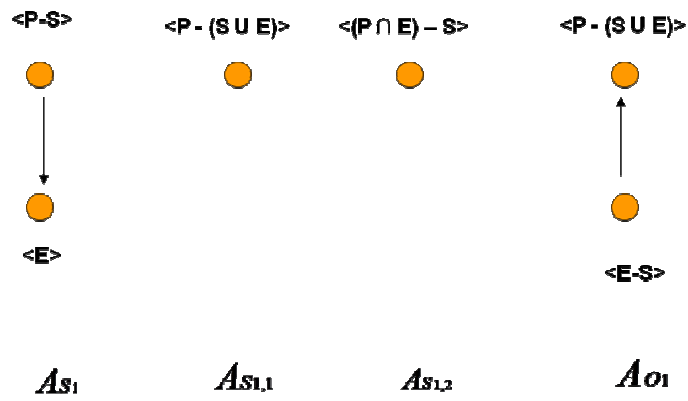


Figure 5.  $A_{s1}$  Decomposition -  $A_{o1}$  Construction

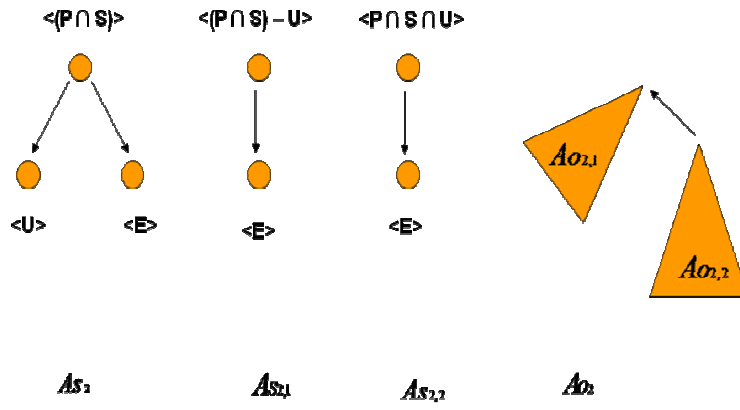


Figure 6.  $A_{S2}$  Decomposition –  $A_{O2}$  Definition

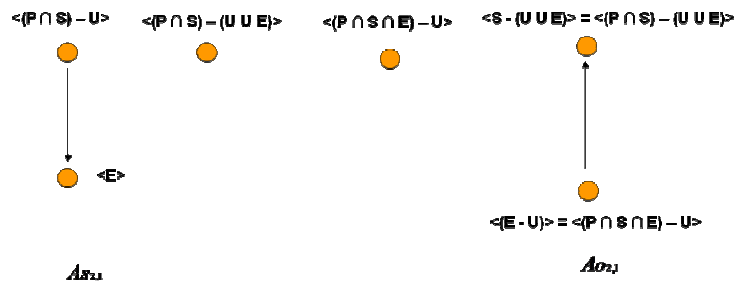


Figure 7.  $A_{S2,1}$  Decomposition –  $A_{O2,1}$  construction

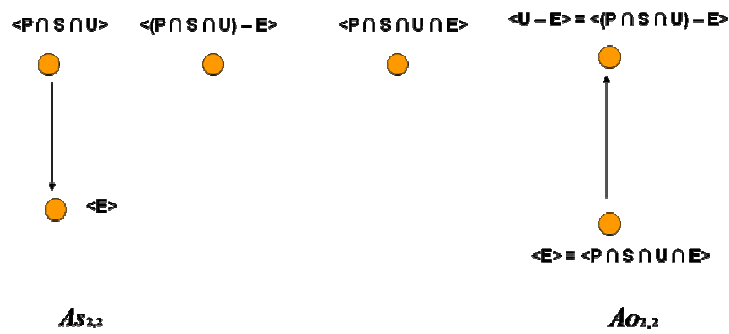


Figure 8.  $A_{S2,2}$  Decomposition –  $A_{O2,2}$  construction



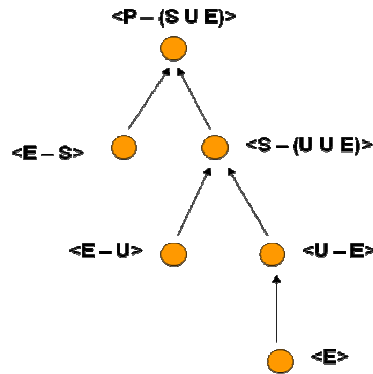


Figure 9. Construction of object classes

The specialization hierarchy  $A_s$  is defined as follows:

$$\langle S \rangle \text{ is-a } \langle P \rangle; \langle E \rangle \text{ is-a } \langle P \rangle; \langle U \rangle \text{ is-a } \langle S \rangle.$$

The root objects are partitioned step by step with respect to the left child. The root labels use the set theory operators to reflect the partitioning. Merge is a recursive procedure which allows the composition of the disjoint classes until this procedure results in the object specialization hierarchy  $A_o$ . This specialization hierarchy is made up of nine nodes, i.e., the algorithm provides nine disjoint classes.

In order to prove the *class completeness*, in [9] the intersection of an object class with a semantic class is either the object class or the empty class. In our example, this can be verified directly. The algorithm of maximum step generates the maximum number of object classes ensuring the class completeness, and generates the maximum number of intermediate semantic classes useful for achieving the *concept completeness*. Different partitioning algorithms can be exploited to obtain the disjoint classes. Analogously, different equivalent graphs of object classes can be obtained starting from the disjoint classes [16].

In order to show how a graph of object classes limits the flexibility in modelling the changes occurring in real life, let us suppose that the university student John becomes an employee. In this case the corresponding object instance must be removed from class  $\langle P \cap S \cap U \rangle - E \rangle$  and must be inserted into class  $\langle P \cap S \cap U \cap E \rangle$ . See the two classes  $\langle U - E \rangle$  and  $\langle E \rangle$  in Figures 8 and 9. If later on John completes his studies, the corresponding object instance must be removed from class  $\langle P \cap S \cap U \cap E \rangle$  and must be inserted into class  $\langle P \cap S \cap E \rangle - S \rangle$ . See the two classes  $\langle E \rangle$  and  $\langle E - S \rangle$  in Figure 9. On the other hand, in the graph of semantic classes, the object instance corresponding to John is inserted into class  $\langle E \rangle$  when the student John becomes an employee and is removed from class  $\langle U \rangle$  when the student John completes his studies. See these classes in the specialization hierarchy  $A_s$  of Figure 3.

In order to complete this section, it is useful to observe that the number of object classes resulting from a partitioning algorithm is related to the graph structure of the initial semantic classes. Its value ranges from the single path tree, i.e., a tree with the maximum depth, where each node has at most one direct descendent, to a tree with the minimum depth, i.e., a single level specialization tree, where all the root descendents

are direct descendents of the root. The number of object classes is  $n+1$ , in the case of the single path tree, and is  $2^n$  in the case of the single level specialization hierarchy. In Figure 2, for example, the number of disjoint classes is 6, as shown in Figure 8 where the disjoint classes are recomposed into a graph. The number of disjoint classes resulting from a single path tree defined by four classes is 5, whereas the number of disjoint classes resulting from the single level specialization hierarchy is  $2^4 = 16$ . The single level specialization hierarchy is the concept structure that generates the maximum number of disjoint classes, i.e., the maximum number of object classes. Each class can be understood as a box with a label. The boxes are potentialities that can be either occupied or empty.

### 1.3. Items Connected to a Concept

There are several basic items connected to a concept, and one possible way to locate them is as follows: a *term* is a linguistic entity. It *denotes* things and *connotes* a concept. A concept, in turn, has an *extension* and an *intension*. The extension of a concept is a *set* /a *class* of all those things that *fall under* the concept. There may be many different terms which denote the same things but connote different concepts. That is, these different concepts have the same extension but they differ in their intension. By an intension of a concept we mean something which we have to “understand” or “grasp” in order to correctly use the concept in question. Hence, we may say that, the intension of concept is the knowledge content that is required in order to recognize a thing belonging to the extension of the concept in question [5, 14].

The relations between concepts enable us to make conceptual structures. The basic relation between concepts is an intensional containment relation [4], [5], [14] and it is this intensional containment relation between concepts, which we call the *is-in* relation. More formally, let there be two concepts  $u$  and  $v$ . When a concept  $u$  contains intensionally concept  $v$ , we may say that the intension of concept  $u$  contains the intension of a concept  $v$  or that the intension of the concept  $u$  involves the intension of concept  $v$ . This involves intensional containment relation is denoted as follows:  $u \geq v$ . The transition from intensions to extensions reverses the containment relation, i.e., the intensional containment relation between concepts  $u$  and  $v$  is converse to the extensional set-theoretical subset-relation between their extensions. Thus, we get,

$$\begin{array}{c} u \geq v \\ \downarrow \\ \langle U \subseteq V \rangle \end{array}$$

where  $U$  and  $V$  are the extensions of the concepts  $u$  and  $v$ , respectively. For example, if the concept of a dog contains intensionally the concept of a quadruped, then the extension of the concept of the quadruped, i.e., the set of four-footed animals, contains extensionally as a subset the extension of the concept of the dog, i.e., the set of dogs. Observe, though, that we can deduce from concepts to their extensions, i.e., sets, but not conversely, because for every set there may be many different concepts, whose extension that set is.

Based on the *intensional inclusion* relation, the following relations of *compatibility*  $\perp$ , *uncompatibility*  $\top$ , *comparability*  $H$ , *uncomparability*  $I$  and *intensional restricted negation*  $\neg^r$  are introduced [11]. Concept *student* is compatible with concept *person* since  $(\exists x)(x \geq \textit{student} \wedge x \geq \textit{person})$ ; concept *student* is incompatible with concept *professor* since  $\neg(\exists x)(x \geq \textit{student} \wedge x \geq \textit{professor})$ , concept *student* is comparable with

concept *employee* since  $(\exists x)(student \geq x \wedge employee \geq x)$ . Properties of these relations allow to define concept constructors, which have correspondence with the set-theory partitioning operators. If  $u$  is *compatible* with  $v$ , the least upper bound, denoted by  $\oplus$  exists. Thus, the concept *person*  $\oplus$  *student* can be defined. Correspondently, at set theoretical level, the following intersection of classes  $\langle person \rangle \cap \langle student \rangle$  is obtained. If  $u$  is *not compatible* with  $v$ , then the two concepts  $u$  and  $v$  are incompatible. Correspondently, at the set theory level disjoint classes are defined. If  $u$  is *comparable* with  $v$ , the greatest lower bound, denoted by  $\otimes$  exists; thus, the concept *student*  $\otimes$  *employee* can be defined. Correspondently, at set theoretical level a class is obtained as union of the following classes:  $\langle person \rangle \cup \langle employee \rangle$ .

In concept theory, intensional negation is problematic but in the case of concepts corresponding to graphs of *semantic classes*, the concept corresponding to the root node of the graph can be taken as the universe of discourse and a restricted intensional inclusion relation can be considered. This allows to introduce the  $\Phi$  operator as follows: *person*  $\Phi$  *student* = *person*  $\oplus$   $\neg^r$ *student*. Correspondently, at set theoretical level, the following class is defined:  $\langle person \rangle - \langle student \rangle$ .

## 2. The methodology

An algorithm to define concept structures related to class of objects is proposed. In Section 2.1 concepts and concepts operators corresponding to classes and partitioning operators of classes are introduced. In section 2.2, the concept structure on which the algorithm works is defined. In section 2.3 the algorithm is proposed.

### 2.1. Formality

In section 1.3, concepts were introduced using single labels, for example concept  $u$ . However, as a class is defined through a set of objects and a set of attributes, in the following, concepts are introduced through information contents. Information content is a primitive, undefined notion. Our approach characterizes a concept through a finite set of information contents, which represent attributes of a class. The following definitions are given:

- An information content  $u_j$  is a concept.
  - A  $u$  is a concept  $\Leftrightarrow \exists$  information contents  $u_j / u \geq u_j$
  - $u = [u_j, j \in J]$
- ↓
- $\langle U, \{ u_j, j \in J^* \} \rangle$

$J$  is a finite index set.  $U$  is the set of class objects and  $\langle \{ u_j, j \in J^* \} \rangle$  is the set of class attributes. Let us explicitly observe that indexes are introduced to define general algorithms. For specific examples, indexed information contents can be substituted by different terms. Examples of information contents are the attributes *name*, *income*. The concept *person* requires the existence of information contents, in our example *person* = [*name*, *income*]. The following properties hold:

- $[u_j, j \in J] \geq [v_i, i \in I] \Rightarrow \forall i \in I, \exists! j \in J / u_j \geq v_i, I \subseteq J.$
- $[u_j, j \in J] \geq [v_i, i \in I] \Rightarrow [u_j, j \in J] = [v_i, u_j, i \in I, j \in J^*] J^*=J-I.$

This latter property states that if concept  $[u_j, j \in J]$  has more information content than concept  $[v_i, i \in I]$ , thus the information content  $[u_j, j \in J^*]$  is *specific* to the concept  $[u_j, j \in J]$ . For example,  
 $[name\ income\ matriculation] \geq [name\ income] \Rightarrow [matriculation]$  is specific to concept  $[name\ income\ matriculation]$ .

- $[u_j, j \in J] \geq [v_i, i \in I] \Rightarrow [u_j, j \in J] \geq [v_i, i \in I]$   
 $\downarrow$   
 $\langle U \subseteq V \rangle$

the above property, called *duality property*, states that more intension corresponds to less extension and vice-versa.

- $[u_j, j \in J] \geq [v_i, i \in I] \Rightarrow [u_j, j \in J] \oplus [v_i, i \in I] = [u_j, j \in J]$
- $[u_j, j \in J] \geq [v_i, i \in I] \Rightarrow [u_j, j \in J] \otimes [v_i, i \in I] = [v_i, i \in I].$

the above properties, called *intensional sum and intensional product*, can be generalized as follows:

- $[u_j, j \in J] \perp [v_i, i \in I], \Rightarrow [u_j, j \in J] \oplus [v_i, i \in I]$   
 $\downarrow$   
 $\langle U \cap V \rangle$
- $[u_j, j \in J] \top [v_i, i \in I] \Rightarrow [u_j, j \in J] \oplus [v_i, i \in I]$   
 $\downarrow$   
 $\langle U \cap V \rangle = \emptyset$
- $[u_j, j \in J] \text{H} [v_i, i \in I], \Rightarrow [u_j, j \in J] \otimes [v_i, i \in I]$   
 $\downarrow$   
 $\langle U \cup V \rangle$
- $[u_j, j \in J] \geq [v_i, i \in I]$   
 $\downarrow$   
 $\langle U, \{ u_j, j \in J^* \} \rangle \text{ is-a } \langle V, \{ v_i, i \in I \} \rangle$

An information content of a concept can be restricted to a given concept. As an example, let us consider the following information contents restricted to the concept *person*:  $[name|_p\ income|_p]$  where  $name|_p$  is the information content of the concept *name* restricted to the concept *person* denoted by *p*, and  $income|_p$  is the information content of concept *income* restricted to the concept *person* denoted by *p*. In the following a property and an example are given:

- $[u_j, j \in J] \geq [v_i, i \in I]$  and  $a \geq b$ , then  $[u_{j \wedge a}, j \in J] \geq [v_{i \wedge b}, i \in I]$ .

$[name\ income\ matriculation] \geq [name\ income]$  and  $student \geq person \Rightarrow [name|_s\ income|_s\ matriculation|_s] \geq [name|_p\ income|_p]$ . Concept  $[name|_p\ income|_p]$  is intensionally contained in concept  $[name|_s\ income|_s\ matriculation|_s]$ . Concept  $matriculation|_s$  is specific to the concept  $[name|_s\ income|_s\ matriculation|_s]$ .

In the following, the two operators  $\oplus$  and  $\Phi$  are applied to concepts defined through restricted information contents.

- $[u_{j \wedge u}, j \in J] \oplus [v_{i \wedge v}, i \in I] = [u_{j \wedge u \oplus v}, j \in J]$   
 $\downarrow$   
 $\langle (V \cap U), \{v_i, i \in I, u_j, j \in J^*\} \rangle$
- $[u_{j \wedge u}, j \in J] \Phi [v_{i \wedge v}, i \in I] = [v_{i \wedge u \Phi v}, i \in I]$   
 $\downarrow$   
 $\langle (V - U), \{v_i, i \in I\} \rangle$

Concepts can be organized into structures defined as follows:

*concept structure*  $\Leftrightarrow$  • the nodes are concepts;  
 • the links are intensional containment relations between concepts.

The *incompatibility* relation is exploited to relate the intensional and extensional aspects of concepts.

$$\begin{aligned} & [v_{i \wedge u \Phi v}, i \in I] \text{ T } [u_{j \wedge u}, j \in J] \\ & \downarrow \\ & \langle U, \{u_j, j \in J^*\} \rangle \cap \langle (V - U), \{v_i, i \in I\} \rangle = \emptyset \end{aligned}$$

The proposed formalism is sufficient to define the subsequent phases of the methodology.

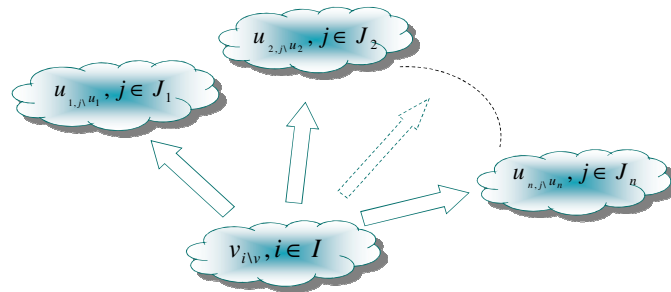
## 2.2. Initial concept structure

In this section, the initial concept structure, i.e., the concept structure to which the algorithm of concept construction must be applied is introduced.

A graph of semantic classes can be *correctly* transformed into a *single level specialization hierarchy*, i.e. into a graph of semantic classes with only direct descendents of the root. The graph in Figure 2, for example, can be correctly transformed into a single level specialization hierarchy with the root *person* and the three direct descendents  $\langle student \rangle$ ,  $\langle employee \rangle$  and  $\langle univ\_student \rangle$ . In addition by transitivity, the objects of the *univ\_student* class are enclosed within the objects of the *person* class. In the single level specialization hierarchy, the class *Univ\_student* has the

attributes from classes *person* and *student* and has also its own specific attribute *matriculation*.

A single level specialization hierarchy of classes can be correctly transformed into a single level generalization hierarchy of concepts. This is achieved by establishing a one-to-one correspondence between classes and concepts. The single level generalization hierarchy is called *initial concept structure*. This structure is shown in Figure 10.



**Figure 10.** Initial concept structure

- The *initial concept structure* is the most general concept structure in which a graph of semantic classes can be transformed, i.e., the structure that with respect to the original graph of semantic classes generates the maximum number of object classes and the maximum number of intermediate classes.
- The *general concept*  $[v_{i \setminus V}, i \in I]$ , corresponds to the root label of the semantic classes graph, i.e.  $\langle V, \{v_i, i \in I\} \rangle$
- The *basic concepts*  $u_1 = [u_{1,j}u_1, j \in J_1]$  .....  $u_n = [u_{n,j}u_n, j \in J_n]$  correspond to the concepts of the remaining graph nodes of the specialization hierarchy.
- The general concept is *compatible* with each basic concept. Thus:  
 $[u_{k,j}u_k, j \in J_k] \geq [v_{i \setminus V}, i \in I]$

### 2.3. The algorithm

The approach to design the concept structures related to classes of objects consists in constructing and merging a finite number of generalization hierarchies of concepts, one for each basic concept of the initial concept structure. All the generalization hierarchies

result in the same leaves which are also the leaves of the designed concept structure. These leaves define incompatible concepts that can be mapped to disjoint classes.

A one-to-one correspondence holds between a partitioning *algorithm of maximum steps* and a corresponding algorithm to construct a generalization hierarchy of concepts.

For each partitioning algorithm applied to a graph of semantic classes, it is possible to determine an *algorithm of maximum steps* applied to the corresponding single level specialization hierarchy. This encloses all the intermediate classes and all the disjoint classes resulting from the considered algorithm. Thus, a corresponding algorithm of concept construction which define the maximum number of concepts and the maximum number of intensional inclusion relations to guarantee *concept completeness* can be introduced.

An algorithm, called the *Concept Construction Algorithm*, corresponding to the partitioning algorithm of maximum step is proposed to design a concept structure:

**Concept Construction Algorithm ( $\theta_i$ )**

**Begin**

If  $\theta_1$  then

Return  $R_{\theta_1}$

Else

If  $\theta_2$  then

Return  $R_{\theta_2}$

Else begin

Decompose  $\theta_i$  in  $\theta_{i1}$  and  $\theta_{i2}$

$R_{\theta_{i1}} =$  Concept Construction Algorithm ( $\theta_{i1}$ )

$R_{\theta_{i2}} =$  Concept Construction Algorithm ( $\theta_{i2}$ )

$R_{\theta_i} =$  Merge ( $R_{\theta_{i1}}, R_{\theta_{i2}}$ )

Return  $R_{\theta_i}$

**End**

The above algorithm is based on the direct solution of the problem applied to the initial concept structures  $\theta_1$  and  $\theta_2$ :

$$\theta_1 = [u_{j \in J}, j \in J] \geq [v_i, i \in I]$$

$$\theta_2 = [u_{1, j \in J_1}, j \in J_1] \geq [v_i, i \in I]$$

$$[u_{2, j \in J_2}, j \in J_2] \geq [v_i, i \in I]$$

$R_{\theta_1}$  is defined by only a generalization hierarchy enclosing 3 concepts;  $R_{\theta_2}$  is defined by two merged generalization hierarchies, one for each basic concept.  $R_{\theta_2}$  encloses nine concepts. The four leaves of this concept structure can be mapped to a graph of object classes. The concept construction approach is defined recursively for the general initial concept structure  $\theta_i = [u_{i,j}u_i, j \in J_i]_{i=\{1, \dots, n\}} \geq [v_i, i \in I]$ . The leaves of each generalization hierarchy define incompatible concepts which can be mapped to disjoint classes. This is sufficient to establish a correct link from the intensional to the extensional level of concepts. The disjoint classes can be organized into graphs of object classes. Many graphs of object classes can be associated with disjoint classes; these graphs define an equivalence class of logical database models. The leaves of the resulting concept structure can be mapped to a representative graph of the object classes. Likewise, many graphs of semantic classes can be associated with the graph representative of the object classes. All of them define an equivalence class of graphs. A one-to-one correspondence holds between a graph that is representative of the object classes and a graph that is representative of the semantic classes. As an example, let us consider the initial concept structure:

$$\theta_2$$

$$[name|_s \ income|_s \ matriculation|_s] \geq [name|_p \ income|_p]$$

$$[name|_e \ income|_e \ salary|_e] \geq [name|_p \ income|_p]$$

The resulting concept structure  $R_{\theta_2}$  is shown in Figure 11. In this figure all and only the concepts related to databases classes are preserved, as well as all and only the logical implications among them.

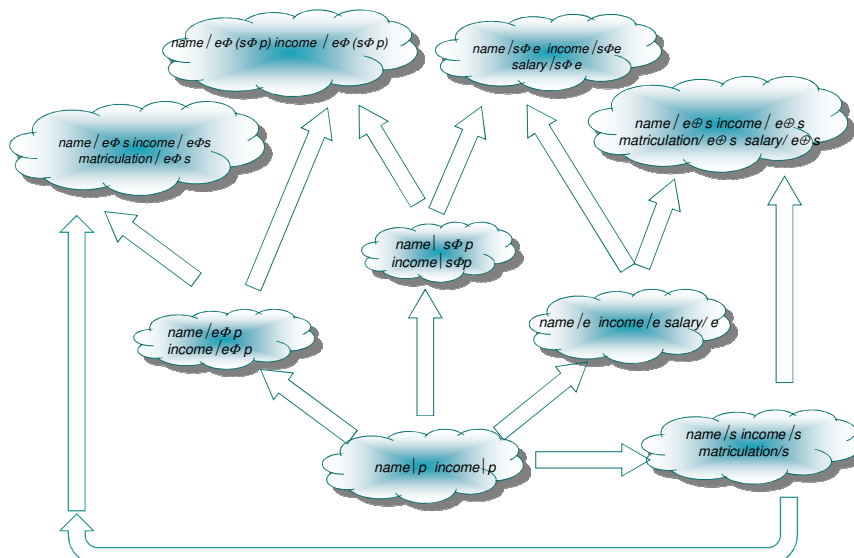
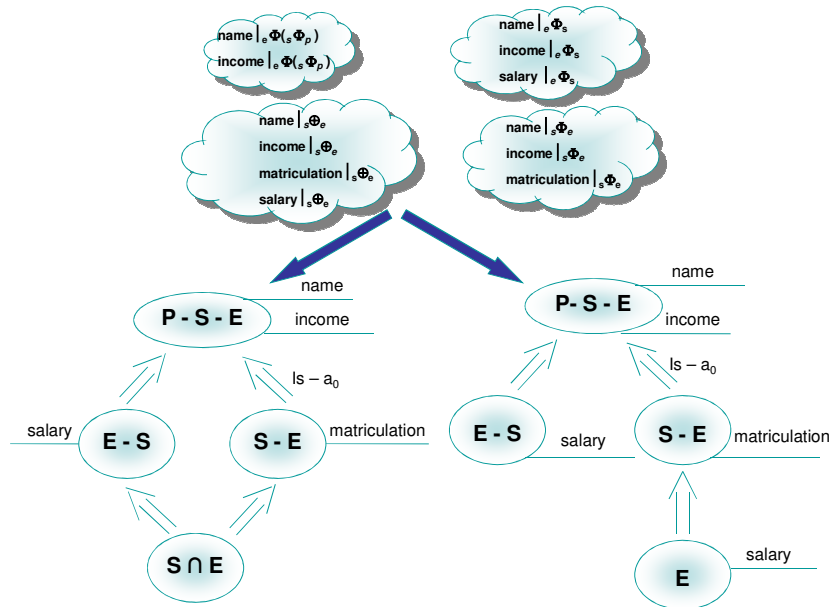


Figure 11. Resulting Concept Structure





**Figure 12.** Intensional and Extensional Aspects of Concepts

The leaves of the resulting concept structure define incompatible concepts which are mapped to disjoint classes. In Figure 12 two of the possible three graphs of object classes are represented and the link between the intensional and extensional aspects of concepts is represented. In turn object classes can be mapped to semantic classes. Figures 11 and 12 show that the intensional and extensional levels are two really different levels; the former is defined by nine concepts, the latter is defined by four classes.

### 3. Discussion

There are several reasons to separate intensional concept theory from extensional set theory [14]. For instance: i) intensions determine extensions, but not conversely, ii) whether a thing belongs to a set is decided primarily by intension, iii) a concept can be used meaningfully even when there are not, nor will there ever will be, any individuals belonging to the extension of the concept in question, iv) there can be many non-identical but co-extensional concepts, v) an extension of a concept may vary according to the context, and vi) from Gödel's two Incompleteness Theorems it follows that intensions cannot be wholly eliminated from set theory.

In usual conceptual database modeling and software engineering applications, it is supposed that a one-one correspondence between the extension and the intension of concepts holds. This means that the intensional concept level and the extensional set-theoretical level are collapsed into a common level and no longer distinguished. The methodology proposed in this paper relates concept theory to computer science distinguishing the concept level from the set-theoretical level. The proposed

methodology allows to solve the following database preservation problem [8]: Introduce information contents and intensional constructors able to:

- Define a general concept  $[v_{i \setminus v}, i \in I]$ .
- Determine an *initial concept structure* within generalization hierarchies of concepts.
- Define an algorithm *to construct a concept structure* that:
  - Results in leaves which are incompatible concepts.
  - Encloses all and only the concepts related to the initial concept structure.
  - Encloses all and only the intensional inclusion relations between concepts.
- Define a *mapping* from the resulting concept structure to an acyclic graph/a specialization hierarchy of classes supported by object database systems.

This problem can be approached and solved not only for traditional databases but also for textual/image databases. Moreover, the definition of concepts in terms of information contents is flexible enough to model a class attribute as well as a textual/visual term. In these cases, the problem can be specifically approached. A negative aspect is the algorithm complexity: let us say  $n$  the number of basic concepts of the initial concept structure, the algorithm complexity is  $n2^n$ . However, it is important to observe that if the original graph of semantic classes encloses necessarily empty intersections of classes, the complexity of a partitioning algorithm can be consistently reduced. A significant example is the case in which the directed descendents of the root have all empty intersections. In this case, many systems define a special top root class for each database and the schema is populated by specialization of this class [9]. Further, the object classes are really created only when at least one element belongs to the class. At the concept level, the general concept  $[v_{i \setminus v}, i \in I]$  corresponding to the special top root class can be disregarded and the concept construction algorithm can be applied separately to each of the  $n$  initial concept structures corresponding to the  $n$  independent subtrees.

#### 4. Conclusions and further developments

An algorithm is introduced to design concept structures related to object classes/categories supported by computer systems. Although concept theory has a formal background, these algorithms are not yet available. The approach is supported by a *methodology* which starts from algorithms of object decomposition proposed in computer science and reaches an algorithm of concept construction related to class/categories of objects. A negative aspect is the algorithm complexity. The paper evidences cases in which the complexity can be consistently reduced.

Using an *is-a* relation, objects to be modeled are presupposed to exist, whereas using an *is-in* relation, the existence of objects falling under those concepts is not presupposed. This difference is crucial for example when we are designing an object which does not yet exist but about which we have plenty of conceptual information

and of which we must build a conceptual model. New applications which makes use of a reduced number of concepts are under development.

### Acknowledgments

This paper derives from a research undertaken by the authors during periods of the following CNR Short Term Mobility Programs: Pori-2009/ Pisa-2010. ISTI-CNR-Pisa and Tampere University of Technology/Pori approve the publication.

Figures 1, 11 and 12 have been reprinted from figures 1, 3 and 4 of the “Ontology for Database Preservation” by Elvira Locuratolo and Jari Palomäki in the book “Ontology-Based Applications for Enterprise Systems and Knowledge Management” edited by Mohammad Nazir Ahmad, Robert M. Colomb & Mohd Syazwan Abdullah, pp. 141-157, Copyright 2013, with kind permission from IGI-Global.

The authors wish to thank the husband of Dr. Locuratolo, Mr. Antonio Canonico, for his help during the preparation of this paper.

### References:

- [1] A. Badia, From Conceptual Models to Data Models, In: P. Van Bommel (Ed.), Transformation of Knowledge, Information and Data: Theory and Applications, IGI Global, Hershey, London, 2005, pp. 148-170.
- [2] A. F. Cardenas, A. F., D. McLeod, Research Foundations in Object-Oriented and Semantic Database Systems. Prentice Hall, Englewood Cliffs, NJ 07632, 1990.
- [3] R. Elmasri, R., S. Navathe, Fundamental of Database System, Addison-Wesley, 2000.
- [4] R. Kauppi, Einführung in die Theorie der Begriffssysteme, Acta Universitatis Tamperensis, Ser. A., 15 University of Tampere, Tampere, 1967.
- [5] H. Kangassalo, COMIC: A system and methodology for conceptual modeling and information construction, Data and Knowledge Engineering 9, (1992/93) pp. 287-319.
- [6] H. Kangassalo, Approaches to the Active Conceptual Modeling of Learning, in P.P. Chen, L.Y. Wong (Eds.), Active Conceptual Modeling of Learning, Berlin, Springer-Verlag, 2007, pp. 168-193.
- [7] E. Locuratolo, ASSO: Portability as a Methodological Goal, *Technical Report IEI B4-05-02* <http://puma.isti.cnr.it/linkdoc.php?idauth=21&idcol=1&icode=1998-TR-003&authority=cnr.iei&collection=cnr.isti&langver=it>
- [8] E. I. Locuratolo, J. Palomäki, [Ontology for database preservation](#), In: Ontology-based applications for Enterprise Systems and Knowledge Management.

Mohammad Nazir Ahmad, Robert M. Colomb, Mohd Syazwan Abdullah (eds.).  
USA: IGI Global, 2013, pp. 141 - 157.

- [9] E. Locuratolo, F. Rabitti, Conceptual Classes and System Classes in Object Databases, *Acta Informatica* 35(1998) 181-210.
- [10] E. Locuratolo, J. Palomäki, Extensional and Intensional Aspects of Conceptual Design, in: H. Jaakkola, Y. Kiyoki, T. Tokuda (Eds) *Information Modelling and Knowledge Bases XIX*, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, 2008, pp.160-169.
- [11] E. Locuratolo, J. Palomäki, Perspective for Database Preservation, In: *Encyclopedia of Information Science and Technology- Third Edition*, Idea Group Publishing, 2015, in print. DOI:10.4018/978-1-4666-5888-2.
- [12] B. Nebel, G. Smolka, Representation and Reasoning with Attributive Descriptions, in: K. H. Bläsius, U. Hedstück, C. R. Rollinger (Eds), *Sorts and Types in Artificial Intelligence*. Berlin, Springer-Verlag, 1990, pp. 112-139.
- [13] B. Nixon, J. Mylopoulos, Integration Issue in Implementing Semantic Data Models, in: F. Bancilhon, P. Bueman (Eds.), *Advances in Database Programming Languages*, ACM Press, 1990, pp. 187-217.
- [14] J. Palomäki, *From Concepts to Concept Theory*, *Acta Universitatis Tamperensis*, Ser. A. 416, University of Tampere, Tampere, 1994.
- [15] J.M. Petit, M.S. Hacid, From Conceptual Database Schemas to Logical Database Tuning, in: P. Van Bommel (Ed.), *Transformation of Knowledge, Information and Data: Theory and Applications*, IGI Global, Hershey, London, 2005, pp 52-74.
- [16] A. M. Spagnolo, *Incrementare la Qualità in Ambito Basi di Dati*, Università degli Studi di Pisa, Tesi, 2000.

# Visualization of Ontologies on the Basis of Cognitive Frames for Knowledge Transmission

Pavel LOMOV<sup>a</sup> and Maxim SHISHAEV<sup>b</sup>

<sup>a</sup>*Institute for Informatics of Kola Science Center of RAS*

<sup>b</sup>*Kola Branch of Petrozavodsk State University*

**Abstract.** In this work the ontologies visualization technology, focused first of all on simplification of getting knowledge from them by the expert is offered. For this purpose it is proposed to form for concepts of ontology special structures – cognitive frames. Each cognitive frame includes the build in a special way fragment of ontology and the visual image, corresponding to it. It is expected that showing cognitive frames for a concept during visualization instead of just showing any terms linked with it will be more useful for presenting of the concept's meaning. In this paper, we consider only the forming the content of cognitive frames based on common relationships from the upper-level ontologies such as "taxonomy", "partonomy" and "dependence". We also provide experiment evaluating the cognitive qualities of frames created for the concepts of application ontology.

**Keywords.** ontology visualization, semantic web, ontology comprehension, cognitive frame.

## Introduction

Visualization of ontologies is an important aspect of their practical use. Good visualization ensures adequate comprehension of ontology or its fragments by experts [1] in context of various tasks of knowledge engineering. Effectiveness of a particular approach to the visualization of the ontology depends essentially on the task to be solved.

One of the tasks requiring ontology visual representation is a sensemaking [2]. It consists in understanding by the user the common structure of the ontology leaving insignificant specific details. This problem usually occurs when you try to reuse the ontology. In this case, the process of sensemaking allows the user to decide whether a given ontology or its fragment is suitable, in general, for particular application. Technology and software tools aimed to solve this problem are presented in the papers [2, 3-5]. Their distinctive feature is the scoping on building high level overviews of ontology, zooming and filtering the displayed items.

Another important issue in the context of formal ontology visualization is to visualize the results of logical deduction. Its essence is to create a visual representation that can illustrate the conclusion of logical statements, and justification of output results [1, 6, 7]. Through this presentation developer can understand in more detail the

ontology, as well as quickly find and correct the problem axioms that lead to semantic conflicts.

Traditionally, within the computer science ontology is used mainly as a means of automated computer processing of knowledge. Today, however, the volume of formalized knowledge contained in the computer's memory becomes comparable to human ones. This opens up opportunities and at the same time raises the problem of efficient transmission of the knowledge contained in the ontology to expert or user. In this case, cognitive qualities of ontology, which are defining how easy and accurate it can be interpreted by an expert to get the meaning of a particular concept, become the most important. The technology investigated in this work focuses on effective transfer of knowledge contained in the ontology to an expert in form of visual images adapted to the interpretation.

Method of ontology visualization proposed in this paper is based on specific structure named *cognitive frame*. In general, the cognitive frame refers to the visualized fragment of ontology, which allows adequately transmitting the knowledge of a target concept to the expert. Adequacy in this case implies a fast and accurate enough for the problem to be solved interpretation of the meaning of the concepts in the context of the mental model of an expert. By their cognitive function frame is close to the notion of viewpoint [8], but unlike the latter, it includes, besides a set of facts about the concept, the corresponding visual image. The requirement for adequate transfer of knowledge from a machine to a human naturally generates a need to consider while forming a cognitive frame the psychological characteristics of a person, the general principles of structuring information by him, as well as some general conceptual framework common to all application ontologies.

According to definition, cognitive frame has two key components - the content corresponding to the ontological context of the target concept, and the visual image, which is presenting to expert. The first component provides an answer to the question *what* should be visualized for effective transition of knowledge on the concept, while the second - *how* to do it. Our approach to human-machine translation of knowledge is based on the observation that the general laws of perception of visual information and the structuring of knowledge by a man are reflected both in his psychological characteristics, and in the description of the concepts in the ontology. Psychological stereotypes of knowledge sensing and interpreting appear in the known principles of Gestalt psychology[9] and in the effect of perceptive stereotypes[10]. On the other hand, within the ontology formed by an expert the concepts are described with implicit or explicit use of relations and meta-concepts invariant to subject areas sourced from a upper-level ontology. Our hypothesis is that following these laws in the process of visualization allow for successful transfer of knowledge contained in ontology to any user.

In this paper, we consider only the question of forming the content of cognitive frames based on common relationships such as "taxonomy", "partonomy" and "dependence". As a universal means for cognitive frame's visualization at this stage of the study a node-link diagram is used. Problem of generation of more complex visual images is supposed to consider in the future.

The paper is organized as follows: section 1 briefly describes the two previous works on the subject. The 2nd section discusses the procedure for forming the content of cognitive frames based on common relations. Section 3 presents the results of an experiment evaluating the cognitive qualities of frames created for the concepts of

application ontology. Final section provides conclusions and directions for further research.

## 1. Background

Visualization of ontologies for knowledge transmission was considered in a series of previous studies by the authors. In paper [11] a technology for automatic generation of simplified modification of OWL-ontology adopted for visualization was proposed. Such modification described in terms of SKOS model [12] is named 'user presentation ontology' (UPO). SKOS model is simpler than OWL model and allows a visual representation as a node-link diagram. To form UPO initial axioms of OWL-ontology represented as a set of elements of SKOS model: concepts, relations and collections. As nodes of obtained graph structure considered concepts corresponding to source OWL-ontology classes. Links represented relationships between OWL-classes.

Next paper [13] was devoted to visualization of UPO, corresponding to some subject ontology, based on cognitive frames. The following general definition of cognitive frame was given:

$$KF(t) = \langle CT, VS \rangle, \quad (1)$$

where  $t$  - target concept of cognitive frame;  $CT$  - content of the frame;  $VS$  - the visual image formed on the basis of the content. Content is a set of links of the form "concept-relation-concept", which are reflecting the meaning of the concept.

The paper also identified the requirements for a cognitive frame:

- Compactness - the frame should contain no more than 7-9 items (according to Miller's "magical number");
- Completeness - the frame is to transfer all the information about the concept;
- Familiarity - visual image frame should be either familiar to the user, or represent the concept from a known viewpoint.

This work also considered the generation of cognitive frames on the basis of invariant for subject domain relations - taxonomy, partonomy and dependence. Further, this type of frame will be called structural cognitive frames. To form their contents appropriate algorithms based on neighborhoods of the target concept have been proposed. Under the  $n$ -neighborhood of a concept is the set of concepts related to the target one by one kind of relationship through  $n-1$  intermediate concept. For example, the concept of  $t$  (Fig. 1) has two neighborhoods on A relation.

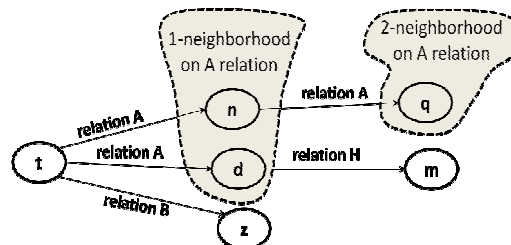


Figure 1. Neighborhoods of concept  $t$ .

When forming the content of structural cognitive frame at each step the notions of only one neighborhood were included. Formation ended by a threshold number of concepts. At the same time to meet the requirement of completeness all concepts of the neighborhood and not part of them is always being added into the content. Along with to avoid transitivity paradoxes appropriate rules is been taken into account when determining the neighborhoods [14].

**2. Improved procedure for the formation of structural cognitive frames**

The procedure of forming the content of structural cognitive frames on the basis of neighborhoods takes into account only one direction of the relationship. This leads to the fact that a structural cognitive frame for a concept could potentially represent its neighbors only to higher and/or lower levels of the hierarchy. For example, partonomy-based cognitive frame for the concept  $t$  will be presenting concepts, which are the parts of  $t$ , and concepts which it is part.

With this forming method are unrepresented concepts, which are at the same hierarchical level as the target one. This does not allow the user to view it in a comparative perspective. For example, if the concept  $t$  is part of the concept  $n$ , then it makes sense to present other parts of  $n$ . This will indicate the distinctive features of the notion of  $t$  compared with other concepts that play the same role for  $n$  (Fig. 2).

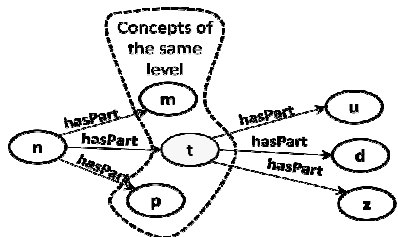


Figure 2. Concepts of the same level of the hierarchy with  $t$  on «hasPart» relation.

To solve this problem and accounting concepts at the same level as the target is further proposed modified procedure of forming the content of structural cognitive frames.

Let introduce the notion of front  $FN_k^f(t)$  (back  $BN_k^f(t)$ )  $n$ -neighborhood of concept  $t$  on  $f$  relation. Recursively define it as a sets of concepts, acting as objects (subject) on  $f$  relation for the concepts of the front or back  $(n-1)$ -neighborhood. Initially, front and back  $0$ -neighborhood for any kind of relations includes only the target concept  $t$  (Fig. 3).

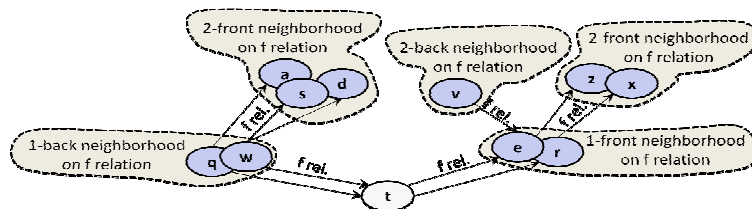
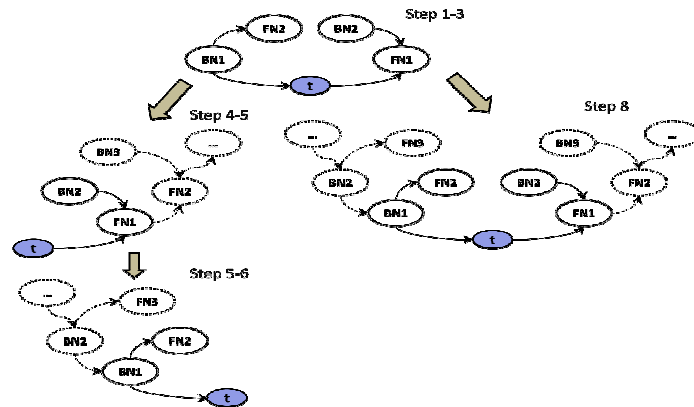


Figure 3. Example of front and back neighborhoods of  $t$  concept.



Modified procedure of forming the content of structural cognitive frame for the target concept  $t$  is the consecutive formation of the front and back neighborhoods and adding them to the content of the frame until the threshold number of concepts. Overall scheme of the procedure is presented in figure 4.



**Figure 4.** The general scheme of the procedure of forming the content of structural cognitive frames.

The modified procedure is as follows:

1. Form  $FN_1^f(t)$  and use it to form a back one neighborhood  $BN_2^f(t)$ ;
2. Form  $BN_1^f(t)$  and use it to form a front one neighborhood  $FN_2^f(t)$ ;
3. Check whether the total number of concepts in neighborhoods formed in step 1 and 2 reached the threshold value. If so, go to step 4. If not, then add the concepts of obtained neighborhoods to content of the frame,  $k = 1$  and go to step 8;
4. Create a separate frame and add to its content neighborhoods obtained in step 1;  $n = 1$  and go to step 5;
5. Building  $FN_{k+1}^f(t)$  on the basis of  $FN_k^f(t)$ . If the frame size threshold is not exceeded, then add  $FN_{k+1}^f(t)$  to the contents of the frame. Then based on the added  $FN_{k+1}^f(t)$  build  $BN_{k+2}^f(t)$  and also trying to add it to the content. If threshold has been reached, then complete the building and go to step 6. If not, then  $k = k + 1$  and repeat step 5;
6. Create a separate frame and add to its content neighborhoods obtained in step 2.  $k = 1$  and go to step 7.
7. Building  $BN_{k+1}^f(t)$  on the basis of  $BN_k^f(t)$ . If the frame size threshold is not exceeded, then add  $BN_{k+1}^f(t)$  to the contents of the frame. Then based on the added  $BN_{k+1}^f(t)$  build  $FN_{k+2}^f(t)$  and also trying to add it to the content. If threshold has been reached, then complete the building and go to step 6. If not, then  $k = k + 1$  and repeat step 7;
8. Building next front-neighborhoods  $FN_{k+1}^f(t)$  on the basis of  $FN_k^f(t)$  obtained in the previous step, and in the case of non-exceedance of the threshold value, add it to the content. Then based on the  $FN_{k+1}^f(t)$  build  $BN_{k+2}^f(t)$  and also trying to add it to the content. Building next back-neighborhoods  $BN_{k+1}^f(t)$  on the basis of  $BN_k^f(t)$  obtained in the previous step, and in the case of non-

exceedance of the threshold value, add it to the content. Then based on the  $BN_{k+1}^f(t)$  build  $FN_{k+2}^f(t)$  and also trying to add it to the content. If this step has been reached the threshold value, the process stops. If not, then  $n = n + 1$ , and repeat step 8.

During this procedure in steps 1-3 the basis of the content of the frame is forming. Next comes it further filling. If the size limit is exceeded at this stage, then for the target concept two separate frames are formed on the basis of the front and back neighborhoods. Note that the formation of neighborhoods complied with the rules of avoiding paradoxes of transitivity considered in work [13].

### 3. Evaluation of technology

Evaluation of the proposed technology was based on the establishment of similarity between automatically generated cognitive frames and sets of facts about the concepts selected by experts from the subject ontology. The rationale for this method of evaluation is that the presence of set of facts formed by the man as a piece of content of the cognitive frame says that he is currently in a similar way interpret the concept. Thus, this assessment shows how some cognitive frame correlates with human ways of structuring information about the objects of reality, and thus is a measure of its cognitivity.

For the experiment five concepts of the ontology of network equipment developed on the basis of a top-level ontology DOLCE [15] were chosen. For each concept structural cognitive frames were formed with use of modified procedure described above.

Five experts were asked to select from the ontology facts relating to each of the chosen concepts and structure them in his discretion. The fact in this case refers to a triple "concept-property-concept". Thus, for each concept experts formed several sets of facts. Further sets of facts generated by experts and cognitive frames were compared. For each frame and set corresponding to one concept determined the count of same facts. Frame and a set of facts with the highest ratings were considered equivalent. For evaluated frame coherence and redundancy with respect to a set of facts was calculated as a proportion of identical and different facts of the total number of facts in the frame.

Averaged results of the experiment are shown in Table 1.

**Table 1.** Evaluation of coherence and redundancy of cognitive frames.

Concept/Cognitive frames	Coherency	Redundancy
<i>Transport layer</i>		
Taxonomy frame	0.55	0.45
Partonomy frame	0.27	0.73
Dependency frame	0.48	0.52
<i>Network router</i>		
Taxonomy frame	0.55	0.45
Partonomy frame	0.77	0.23
Dependency frame	0.75	0.25
<i>Routing protocol OSPF</i>		
Taxonomy frame	0.67	0.33
Partonomy frame	0.60	0.4

Dependency frame	0.50	0.5
<i>Media access control task</i>		
Taxonomy frame	0.77	0.23
Partonomy frame	0.75	0.25
Dependency frame	0.9	0.09
<i>Network interface</i>		
Taxonomy frame	0.69	0.31
Partonomy frame	0.43	0.57
Dependency frame	0.6	0.4

Note that high redundancy due to the fact that not all the experts involved in the number of selected facts describing the concepts of one level of the hierarchy with the target one. This is especially true for hierarchies on taxonomy and partonomy relations. However, after consideration of the relevant frames are automatically constructed, experts noted that it should be made for a complete presentation of the target concepts meaning. For example, it refers to the partonomic frame of “*Transport level*” notion and taxonomic frame for “*Network router*” (fig. 4).

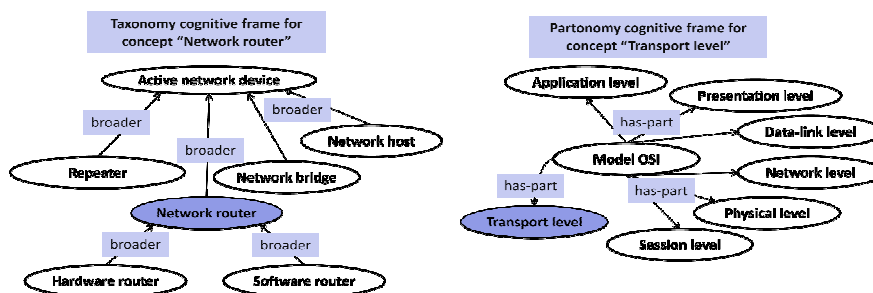


Figure 5. Example of partonomic and taxonomic cognitive frames.

As for the frames based on dependence relation, in most cases, it corresponded to a set of facts, unsuitable for other types of structural frames. This is because the dependency relation is much more specific as compared to partonomy or taxonomy. In this regard, experts combined into one set both facts relevant to this relation and facts corresponding to different specific relations of subject domain. This has led in some instances lower coherence estimation.

In general, the experimental results show sufficient proximity of the frame contents and viewpoints represented by the set of facts selected by experts. This allows us to say about the correctness of the proposed procedure of forming the content, ensuring familiarity and compactness of cognitive frames.

#### 4. Conclusion

This article considers a definition of cognitive frame used as a means of visual representation of the concepts of the ontology. Cognitive frames include a fragment of an ontology that defines a concept, as well as a visual image, which facilitates the interpretation of the concept's meaning. In the next stages of the study is expected to consider different aspects of forming a visual image based on the principles of cognitive computer graphics [16] and Gestalt psychology [9]. Along with this, it is planned to explore the possibility of synthesis of the visualization using standard notations such as IDEF, UML and others.

The presented procedure allow an ontology segmentation for subsequent its representations as cognitive frames. Basis for the formation of such segments is the presence of different hierarchies of ontology concepts for invariant to subject areas relations specified in the top-level ontologies. Using invariant relations, as well as consideration of concepts at different levels of the hierarchy allows to generate the content of structural cognitive frame, satisfying the requirements of compactness, completeness and familiarity for any ontology. In future works it is supposed to determine how to generate a cognitive frame's content considering the target concept as an successor of some concepts of upper-level ontology.

Another important direction of future research is creation of a navigation system for a set of cognitive frames. Such system makes possible to use proposed ontology visualization technology as the basis for user's interface of information systems, oriented on learning and sharing of knowledge between experts.

## References

- [1] J. R. Bergh, "Ontology comprehension", University of Stellenbosch, Master Thesis 2010.
- [2] E. Motta, P. Mulholland, S. Peroni, M. d'Aquin, J. Manuel Gomez-Perez, V. Mendez, F. Zablith A Novel Approach to Visualizing and Navigating Ontologies, *Lecture Notes in Computer Science* Volume 7031, 2011, pp 470-486.
- [3] C. Plaisant, J. Grosjean, B. Bederson, Spacetime: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. *In Proc. of the Intl. Symposium on Information Visualization*, (2002), 57 - 64.
- [4] T. D. Wang, B. Parsia, CropCircles: Topology Sensitive Visualization of OWL Class Hierarchies, *Lecture Notes in Computer Science*, Volume 4273, 2006, pp 695-708.
- [5] B. Shneiderman, Tree Visualization with Tree-Maps: A 2d Space-Filling Approach. *ACM Trans. Graph.*, 1992. 11(1): p. 92-99.
- [6] J. Bauer. *Model exploration to support understanding of ontologies*. Master's thesis, Technische Universität Dresden, 2009.
- [7] T. Liebig, O. Noppens, OntoTrack: Combining browsing and editing with reasoning and explaining for OWL-lite ontologies. *In Proceedings of the 3rd International Semantic Web Conference ISWC 2004*. Hiroshima, Japan. 8-11.
- [8] Acker L., Porter B. Extracting viewpoints from knowledge bases // *In Proceedings of the 12th National Conference on Artificial Intelligence*, 547-552, 1994.
- [9] T.A. Gavrilova, V. A. Gorovoy, E. S. Bolotnikova, Evaluation of the cognitive ergonomics of ontologies on the basis of graph analysis, *Scientific and Technical Information Processing*, December 2010, Volume 37, Issue 6, pp 398-406
- [10] P.N. Johnson-Laird *Mental Models: Towards a cognitive science of language, inference and consciousness*. // Cambridge, VA: Harvard Univ.Press, 1983. 246 p.
- [11] P.A. Lomov, M. G. Shishaev, V. V. Dikovitskiy OWL-ontology transformation for visualization and use as a basis of the user interface, *Scientific magazine "Design Ontology"* - 2012. Samara: Novaya Tehnika, 2012, P. 49-61 ISSN 2223-9537, (in russian).
- [12] SKOS Simple Knowledge Organization System Reference, *W3C Recommendation*, 2009. <http://www.w3.org/TR/skos-reference>.
- [13] P. Lomov, M. Shishaev Technology of Ontology Visualization Based on Cognitive Frames for Graphical User Interface, *Communications in Computer and Information Science*, Volume 394, 2013, pp 54-68, page 54-68. Springer, (2013).
- [14] M. Winston, R. Chaffin, D. Herrmann: A Taxonomy of Part-Whole Relations. *Cognitive Science*, vol.11, pp. 417-444 (1987).
- [15] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, L. Shneider: *WonderWeb. Final Report*. Deliverable D18 (2003).
- [16] V.L. Averbukh, Toward formal definition of conception adequacy in visualization// *Proc. 1997 IEEE Symp. on Visual Languages*, Sept. 23-26, 1997. Isle of Capri, Italy. S. I.: IEEE Comput. Soc. 1997. P. 46-47.

# eLogika – the system for teaching logic

Marek MENŠÍK, Marie DUŽÍ, Jakub GERLICH

VSB – Technical University Ostrava, Department of Computer Science,  
17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic

**Abstract.** In this paper we introduce the Learning Management System (LMS) *eLogika* that has been developed in our department for teaching mathematical logic. There were many reasons that led us to the decision to develop such a system, including inter alia a great amount of students enrolled for the courses on logic. Yet the most important reason was a specific character of logic education. As a result, the *eLogika* system is a web application that provides didactic material for courses on mathematical logic. Its main goal is an automatic test generation and computer-aided test evaluation based on a large database of logic tasks. The system makes it possible to adjust the level of particular tests according to students' knowledge level. To this end we developed a feedback module that makes use of statistics and data mining methods. The system can generate a large number of training as well as exam test variants for each common thematic topic. At the same time it provides effective semi-automatic methods of test rating and evaluation. In the paper we describe particular modules of eLogika with the focus on the modules of data mining and statistics.

**Keywords.** E-learning, logic, data mining, test generation and evaluation

## Introduction

E-learning systems are nowadays broadly used and applied in education, and e-learning has become a predominant form of post-secondary education. In the relatively new LMS market, commercial vendors for corporate and education applications range from new entrants to those that entered the market in the nineties. In addition to commercial packages, many open-source solutions are available. The available systems are usually helpful in the course administration and management, such as online or blended learning, supporting the course materials online, associating students with courses, tracking student performance, storing student submissions, and mediating communication between the students as well as their instructor. Institutions either decide to apply an existing system, such as Moodle, or develop a new system tailored to their own needs. We voted for the latter, because we need a system for teaching and learning mathematical logic. Although the open-source systems typically can be extended by creating plugins for specific new functionalities, eventually we found out that logic has so many specific aspects that it will be easier and more useful to create our own system tailored to teaching and learning mathematical logic.

There are many reasons that led us to this decision. First, the number of students enrolled in the logic courses has been year by year running to three hundred. Due to a great amount of students enrolled in the course on Mathematical logic in our department, a face-to-face individual contact of a tutor and students and their training is almost impossible. Thus it is hardly possible for a teacher and tutors to care of each

student personally. Second, there are students who study in the so-called distant form, and they need to practice logic at home.

Third, we wanted to have an e-learning system tailored to logic education. True, the number of didactic courseware packages oriented to introductory logic courses is quite large. Among them the most elaborate ones are presumably CSLI's courseware packages 'Tarski's Worlds', 'Language, Proof and Logic', 'Hyperproof', 'The Language of First-Order Logic', and 'Turing's World'.<sup>1</sup> Such software packages usually provide sophisticated didactics. However, they all ignore the administration of teaching, like correcting, grading and storing the achieved results. Moreover, they usually provide only a fixed structure for exercises. On the other hand, systems designed to satisfy the administrative requirements (e.g. e-learning systems) are able to deal with the administration of teaching, but they do not reflect the special needs of logic as a discipline. They were designed for those fields of study that have an encyclopaedic character, which is inadequate for logic exercises since the only way of practicing as well as testing logic is *via* test questions. Teaching and studying logic is specific in many aspects. The students do not have to memorize large amount of information. Rather, it is important that they deeply understand the subject so that they are able to actively prove arguments and theorems, check consistency of their assumptions, and/or modify particular proof methods. The students must keep track during the semester in order not to get frustrated by not understanding the subject. If a student is confused or frustrated, he or she may get bored and is unlikely to be motivated to succeed in that class.

Fourth, it is very important that students, as they go through the course, have the possibility to test their knowledge by solving particular tasks, because the most effective way of teaching logic is individual logic-problem solving. Though solving examples is the best way to learn modern logic and become familiar with logic notation, there is a deplorable lack of relevant literature. Textbooks with sample solutions and hints for correct answers are very rare, if not entirely missing. Last but not least, the teachers need to test students' knowledge not only at the final exams but also during the semester in order to be able to adjust teaching methods in accordance with students needs. In general, we also wanted to motivate students to learning logic, which is usually not a particularly favoured subject. Thus we decided to develop an LMS system for mathematical logic that would incorporate specific functionality for logic, including gamification such as logic puzzles and rewards for successful or innovative solutions, and so like. As a result, we have been developing an *eLogika* system that we are going to describe below.

In [1] and [2] we briefly introduced two e-learning systems, to wit LMS Organon and eLogika. The *LMS ORGANON* has been developed in the Faculty of Philosophy of the University of West Bohemia in Pilsner, and it is primarily designed for introductory logic courses. It is able to accommodate logic symbolism as well as the other sorts of visualizations like tables or diagrams. Our eLogika system has been designed for more advanced courses on Mathematical logic, and it is particularly tailored to training as well as exam test generation and evaluation. Both the systems satisfy the special needs of logic discipline. The goal of this paper is to describe functionalities of eLogika in more details. We primarily focus here on the feedback modules that apply methods of statistics and data mining.

---

<sup>1</sup> For information on these packages, see, for instance, [10].

The rest of the paper is organized as follows. In Chapter 1 we describe the structure of eLogika system. Chapter 2 introduces the most important modules of the system, to wit test generation module, test evaluation module and the modules of statistics and data mining. eLogika database is described in Chapter 3. The main results are presented in Chapter 4 where we describe modules that provide a feedback, that is statistics and data mining modules. Concluding remarks are contained in Chapter 5.

## 1. The structure of *eLogika*

### 1.1. User-roles in *eLogika*

There are several roles with their specific functions that a user assigned to that role can play. The basic role is that of *Student*. Students can read and study education materials, and perform training as well as exam tests. The students are assigned to particular *seminar* groups, and they have a possibility to follow *lectures*. The seminars are guided by *tutors* and lectures are provided by the *guarantor* of the course. User playing the role of a tutor or guarantor is entitled to insert tasks and exercises to the eLogika database from which particular tests are generated. Additional roles of a guarantor are these. They can specify the conditions for a successful passing the course, that is a minimal and maximal number of credits, the types and number of tests students must execute, and in particular, the guarantor is responsible for the quality of teaching material available online as well as text books. The guarantor also nominates and determines their tutors. The system *administrator* is responsible for system management and its keeping up to date. He/she initiates particular courses in the system, and assigns guarantors and tutors to the courses.

### 1.2. Modular structure of *eLogika*

*eLogika* is a modular system. The most important modules are the module for test generation and management, automatic test evaluation, data mining module, authentication and identification modules. Particular modules are available as *web applications*. Thus it is possible to upgrade the system to user platforms by including plugins that are actually independent of eLogika, because they just call remote web services of eLogika. We voted for this modular structure in the interest of a broader usability and flexibility. The system can be used not only as the entire web application but particular modules can be also run as a mobile application. Moreover, the system is flexible and gradually upgraded. In our Department of Computer Science we made use of this feature also in the education process, because some modules were developed as diploma theses. Currently we are programming the interface for WP8, IOS and Android platforms.

### 1.3. The Added value of the system with respect to a student and a teacher

The system makes it possible to involve students in the learning process indirectly so that they can learn and practice logic wherever and whenever needed. The guarantor of the course inserts into the system particular textbooks, exercises and other education texts. This material is divided into *chapters* and *categories*. The system is monitoring

student's activities, and dependently on which chapter he/she is working on particular training test is generated. After the test evaluation the student is informed about the mistakes they made, including the explication of the reasons why his/her answer is not correct together with a hint to a correct solution. Hence the students do not have to wait for a face-to-face consulting the teacher. Yet the situation is not so simple, which is due to the special character of logic tasks. In case of Yes/No or Wh-questions automatic evaluation works perfectly well. However, many tasks are of a creative character. The student must prove himself a theorem or an argument. In such a case the system can check the correctness of the proof only very roughly, and a manual checking is almost inevitable. Still the system is a great relief both for a teacher and a student. Teacher's capacity is not exhausted by manual checking of those tasks that can be evaluated automatically, and thus they can devote more time to the students who need to consult the way of proving or for some reasons prefer a personal consulting. Moreover, the system provides a feedback on the way students work, what is difficult for them, which are the typical mistakes, and so like. Thus the teacher can tailor and tune education methods individually according to students needs.

#### *1.4. Authentication and identification*

An important aspect of teaching is also checking students' knowledge, of course. Unfortunately, students tend to cheat, which is neither desirable nor fair, and it also misrepresents our feedback on students' skills. It is almost a common practice that a student asks a friend who masters the topic in order to execute the test instead of him/her. If a test is executed in the college, then we can check the identity of students using their ID-cards. However, when tests are executed in a remote site then we need a more reliable method of their identification. To this end we implemented an authentication module that makes use of *biometric information*.

## **2. Brief description of particular modules**

### *2.1. Test generation module*

This module is the module of fundamental system functionality that is test generation and evaluation. Each test is classified into a *type of activity* and a particular *activity*. By the type of activity we mean inter alia a credit test, an exam test, a training test. Particular activity is then for instance the first term of an exam test. As mentioned above, our database of tasks and exercises is divided into topical categories. Similarly, a test is divided into *blocks* and to each block a category of exercises is assigned from which tasks for the test are generated.

Tests can have an *online* character, hence the students execute the test online, or the test is *printed* and the students work using a paper and pen. In the online case each exemplar of a test is unique; hence each student obtains a set of randomly chosen tasks tailored according to the activity and a student. In the printed case the teacher can also vote for unique variants of the test for each student, or for a fixed number of versions. Each printed version of a test has assigned a unique QR code that unambiguously identifies the test. As soon as students obtain their printed versions of the test, each of them must glue his/her QR code uniquely identifying the student. This double identification is necessary for automatic evaluation of tests.



## 2.2. Test evaluation module

The evaluation of tests is executed in two ways. Yes-No and Wh-questions can be evaluated automatically. Creative tasks are checked by tutors manually. The online tests are evaluated as soon as a student ends up his/her work. The printed versions are scanned and loaded into the system. The evaluation module first recognizes the QR code assigned to test as well as the one provided by student. In this way the module assigns the results to that student who worked out the test. The students then obtain a message that the test has been evaluated and they can see and check the results. Yet, though the tests are necessary and useful, automatic test generation and evaluation themselves does not make the system attractive for students. It helps teachers to manage a great amount of students but in general it does not bring out much more new and attractive functionalities. Thus we also implemented a module for statistics and data mining that we are going to describe in the next paragraph.

## 2.3. Statistics and data mining

The eLogika system keeps information about students, their work and executed tests (whether training or exam ones, online or printed) that can be then used for discovering interesting hidden information and dependencies. To this end we have a module of statistics and data mining.

*Statistic* methods are, or should be, an important component of every LMS system. Moreover, the output data of these modules then serve as inputs for data mining methods. Thus we can follow how the students work, analyse which categories of tasks are difficult for them and which are easy, or we can even classify students into groups according to their skills. In this way we obtain a feedback that makes it possible to tune the education methods and generate training tests for the students according to their needs. Another goal of statistic methods is tuning the tests themselves, which is based on the statistic results on the distribution of correct and incorrect answers, as well as the number of attempts to solve the task. Thus we can check whether tests are well prepared, whether particular questions are unambiguously and clearly formulated, and so like.

*Data mining* makes it possible to discover hidden dependencies. In our module we make use of *association rules*, *decision trees* and *cluster* methods. These modules analyse the data obtained by statistical methods like the number of attempts at solving a task, the number of incorrect answers, the distribution of mistakes among particular categories of tasks, and so like. For instance, the association rules point out the fact that in case of only 20% success rate in category  $K_1$  the category  $K_2$  displays similar lack of success. Based on this information the system adjusts the training tests in such a way that the tasks of category  $K_1$  and  $K_2$  are grouped together so that the students can practice them both. In this way we generate training tests that are tailored to the level of skills of a particular group of students. Our experience proved that these training tests are very useful in particular for distant students who do not take part in the regular college seminars.

### 3. eLogika database system

The eLogika system is a complex LMS system that makes it possible to collect data on students' work and their results, questions and answers, the number of attempts, structure of tests, etc. In this chapter we describe data collection and management, the structure of eLogika database, and the methods of data mining in more details.

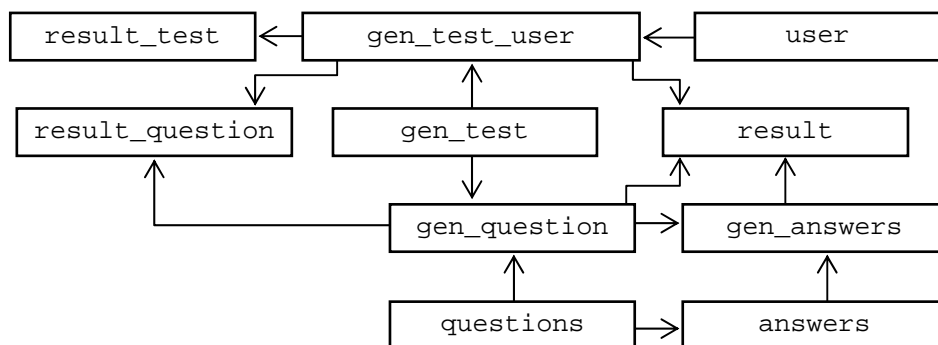
#### 3.1. Data management

In eLogika database we collect a lot of various kinds of data on students, courses, users, etc. For the purpose of test generation and evaluation we need in particular three types of records.

- Data on the *access* to the system; these data can be used for the purpose of optimization of an update of the system. They also serve for grouping the students in accordance with the way of their using the system
- Students' *results* in solving tests and other activities
- Data on particular *questions* and *answers* from which particular tests are generated.

These data are also applicable in order to discover problematic issues in text books and educational materials using the data mining modules.

Figure 1 illustrates a part of eLogika database that contains information about tests.



**Figure 1.** eLogika database structure

The core of the database is the evidence of tasks each of which consist of a *question* and the set of possible *answers* (both correct and incorrect) stored in the tables 'questions' and 'answers', respectively. Each task record (that is both the 'questions' and 'answers' tables) consists of the text formulation (including formulas), reference to the author, assumed time to read and solve the task, date and time of storing into the system. There are two types of questions/tasks. The first type of a question has a set of possible answers assigned to the question, and a student votes for correct ones. These are Wh- or Yes/No questions which are evaluated automatically by the system. Possible answers to each question are stored in the table 'answers'. In training tests it is useful that a student obtains information about the reasons why a given answer is correct or incorrect. To this end we can store in the table 'Answers' also these additional explanations.

A *creative* task does not have prescribed possible answers. A student must create an answer themselves. These are tasks like write the proof steps for a given argument. These tasks are evaluated manually. Creation of a database of admissible answers also to the creative tasks like prove this or that is still work in progress. Currently there is a possibility to store in the table ‘Answers’ just one typical correct answer. Anyway, in our opinion, evaluation of such creative tasks will always be only semi-automatic. We can store a set of admissible proof steps and their compositions, for instance, but a final manual checking will always be necessary.

The system makes it possible to generate test exemplars with particular questions and answers. This is recorded in the table ‘gen\_test’ together with information about the date and time of test execution. Each test exemplar consists of questions (the table ‘gen\_questions’) selected from the table ‘questions’ and selected (correct as well as incorrect) answers (the table ‘gen\_answers’). The table ‘gen\_test\_user’ contains information about particular students who work on the test exemplar. These records are related to students’ results, which in turn are stored in three tables, to wit ‘result’, ‘result\_test’ and ‘result\_question’. The table ‘result’ contains data on questions and answers that the student marked as correct. Records in the table ‘result\_question’ contain the maximal number of credit points that a student can obtain for a correct answer to this question, and the actual number of the points that the student did obtain. The table ‘result\_test’ contains the overall results of particular students in the test. Thus each record has a data on the maximal number of the points obtainable in the test, the actual number of the points that the student obtained, date and time of finishing the test.

Besides these data the system also stores the statistical data like for each task the time needed to solve, the number of changes/corrections, IP address of the user, and so like.

### 3.2. Data warehouse

Data contained in the tables described in paragraph 3.1 are the most important data we need for data mining methods. Yet information contained in these tables need to be pre-processed in order to be used efficiently. For instance, if we wanted to know which answers of a student were not correct, we would have to join six tables, which is certainly not an effective way of providing an answer. First we must extract student’s answers from the table ‘result’, then assign these answers to questions in the table ‘gen\_questions’, compare with correct answers from ‘gen\_answers’, assign to a particular test, etc.

Thus in order to make the complex analyses faster and respond quickly and flexibly we decided to create a data warehouse with a less detailed structure that contains data integrated from multiple source tables.<sup>2</sup> This extraction and pre-processing is usually performed only once, after the end of the exam period. The main source of the data is cleaned, transformed, cataloged and made available for data mining.

In the data warehouse we also maintain data history about the results of past courses and tests. We voted for a typical star schema architecture where the data is arranged into hierarchical groups called dimensions and (aggregate) facts. Dimension tables contain statistic data like the specification of a task, maximum number of credit points and other rather stable attributes common to a greater amount of facts. The

---

<sup>2</sup> For details on data warehouses see, for instance, [3], [4], [5].

tables of facts contain records generated by students and other users during their work with the system. They are for instance test results, questions and user answers, or access records.

Thus the only type of a relationship in the data warehouse is that of the dimension table and the table of facts. The access layer helps users retrieve data. In order to extract data from source tables, transform them, and load them into the data warehouse we created data pumping program. This layer also improves data quality by checking the data, erasing incomplete or inconsistent records. Sometimes it is necessary to adjust or complete missing values of attributes by calculating them. Here we must carefully consider which way is more effective whether to compute and complete missing values when needed, or to pre-process them and store in the warehouse. The former is suitable for frequently asked questions but the database memory is growing. The latter can slow down the response. For instance, in the source database of results there are only those answers that a student voted for. But for the needs of data mining we need complete information about correct and incorrect answers to the respective questions, their weight category, etc. Another example is information about time values. After careful tuning the system we decided to create a special dimension table 'DW\_Time' that contains calendar data like year, month, date, hour and minute, but also aggregated and computed data on academic year, the date in a semester, and so like. This feature makes it possible to quickly filter data and provide a quick response on questions concerning temporal information.

The structure of data warehouse is illustrated by Figure 2. The tables 'DW\_Answer', 'DW\_Test', 'DW\_User', 'DW\_Question', 'DW\_Time' and 'DW\_TestGroup' are dimension tables. They contain basic information about tests and tasks, questions, possible answers and users. The table 'DW\_TestGroup' represents particular date and time of test execution with data like the date of beginning and end, time, capacity, etc. The other tables are fact tables.

The table 'DW\_TestResultQuestion' contains students' results in particular questions. Each record contains a reference to the question, the respective test and user. There are minimal and maximal number of credit points for the task, the actual number of credit points transformed into a relative value within the interval (0,1) and time needed to solve the task. In order to speed up statistic computing, we store here also auxiliary computed values like the square of obtained credit points and time to answer.

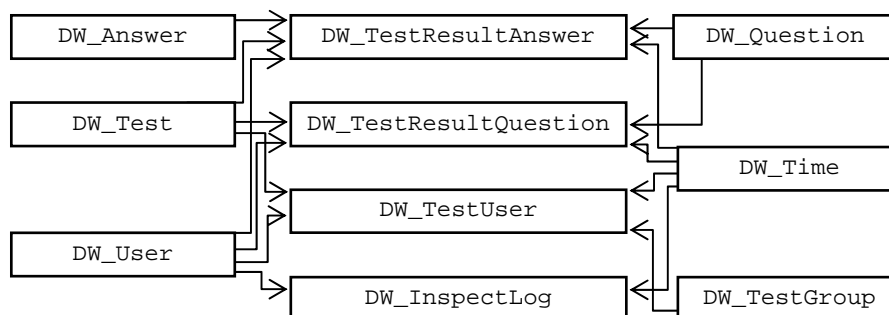


Figure 2. eLogika data warehouse structure

'DW\_TestResultAnswer' contains user answers (both correct and incorrect) to a given question together with information whether a student marked this answer correctly or incorrectly, the number of answer changes, as well as the reference to the dimension table, of course.

'DW\_TestUser' table contains aggregated test results. Hence each record corresponds to one test execution. There are similar attributes here as above together with computed values and references to the respective dimensions.

The 'DW\_InspectLog' table serves to keep track about the access to the system. There are attributes like time of an access, IP address, user, accessed page, used explorer, course, college, etc.

After having pre-processed and transformed data into the warehouse we are ready to use data mining methods.

#### **4. Data-mining and statistics**

The data mining task can be characterized as the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection) and dependencies (association rule mining).<sup>3</sup> The obtained results can serve to prediction or tuning and improvement of our teaching methods.

Now we are going to describe data mining methods applied in the eLogika system.

##### *4.1. Statistics*

Statistic methods are applied in order to classify particular tasks according to their difficulty and success rate in their solving. It is often the case that those tasks that seem easy for us, the teacher, turn out to be difficult for students and vice versa. Or, it can be the case that the specification of the task in the task database is ambiguous or generally not clear enough. Using these methods we discover extremely difficult or extremely easy tasks from the student point of view, and consequently adjust these tasks or generate more training tests so that the students can practice difficult tasks.

When analyzing questions and answers we must specify intervals of success rate and support so that to extract those tasks that exhibit some anomalies and are thus the candidates for improvement. The support is determined by the number of occurrences in particular tests. The greater support, the more accurate the success rate and task evaluation is. We also filter the results according to a semester and course.

As a result we obtain the list of tasks with their success rate and support. Then we examine the list and decide which tasks should be adjusted, corrected, or whether it is desirable to change their specification, and so like.

##### *4.2. Cluster analysis*

Cluster analysis or clustering is the task of grouping data in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each

---

<sup>3</sup> See [http://en.wikipedia.org/wiki/Data\\_mining](http://en.wikipedia.org/wiki/Data_mining), [7], [8].

other than to those in other groups (clusters).<sup>4</sup> Clustering is a main task of exploratory data mining, and a common technique for statistical data analysis. The task of cluster analysis can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Clustering is thus often characterized as a multi-objective optimization problem. It is an iterative process of implicit knowledge discovery that involves trial and error.

#### 4.2.1. Comparing objects

In order to cluster objects we first need to calculate their similarity or distance. The *distance* can be computed only for objects with numeric attributes, because it works with the difference of numeric values and we cannot compute the difference in such values for a text. Or rather, it would be a futile activity. On the other hand the *similarity* is computed for comparison of categorical data.

In our e-learning system we have both categorical and numeric data. For instance, in Table 1 there are data from the record 'DW\_TestUser'.

**Table 1.** Data in the table 'DW\_TestUser'

IdTest	TestNum	IdUser	IdTest Group	Points	Duration	Points Pow	Duration Pow	Try
30	248	286	103	41	5160	1681	26625600	1
30	249	657	103	37	4920	1369	24206400	1
30	251	313	103	13	3120	169	9734400	2
30	252	644	103	16	5160	256	26625600	1

Though all values seem to be numeric, some are categorical. The identifying attributes IdTest, TestNum, IdUser a IdTestGroup have numeric values but we cannot work with them as with numbers. It makes no sense to calculate, for instance, the difference of IdUser values for two users. Numeric values are here number of credit Points, Duration in seconds, and their powers. The attribute Try is a special one. It is the serial number of the attempt to successfully execute the test. In case of an exam test its maximal value is 3. We work with these values both numerically and categorically. For instance, we may want to know the success rate for the first, second and third rate. In this case the Try values are used categorically. On the other hand, when calculating the success rate for a given category of tasks, the Try values should accordingly decrease the number of obtained points, because for instance 90 points in the first attempt can be considered to be more valuable than 90 points in the third attempt. Thus it is a matter of proper tuning particular algorithms for computing *distance* metric and *similarity*.

The distance  $d$  of two vectors  $A$  and  $B$  ( $A, B \in \mathbb{R}^n$ ) is a function  $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  that should satisfy the following conditions.

- Its values are non-negative,  $d(A, B) \geq 0$
- The distance of identical objects is equal to zero,  $d(A, A) = 0$
- It is a symmetric function,  $d(A, B) = d(B, A)$
- Triangle inequality (subadditivity),  $d(A, B) \leq d(A, C) + d(C, B)$ .

<sup>4</sup> See, for instance, [3] and [4].

There are many different definitions of distance, from which we voted for two distances or metrics:

Eukleides distance:

$$d(A, B) = \sqrt{\sum_i (A_i - B_i)^2}$$

Manhattan distance:

$$d(A, B) = \sum_i |A_i - B_i|$$

As mentioned above, in order to determine *similarity* of objects, we must first categorize our data. For instance, the number of credit points that a student can achieve is within the interval (0, 100). Hence without further categorizing these data we would have 100 categories, which is too many. Optimum is usually rather a small number of categories. Students' results are actually categorized by the university study rules. For instance, the European Credit Transfer System (ECTS) classifies students' results into the categories A – F, where A is excellent, B very good, C good, D satisfactory, E sufficient and F means failure. Yet for the purpose of eLogika we needed a bit more detailed categorization and voted for ten categories, each of them of 10 credit points.

Similarly as the distance, similarity can also be calculated using different algorithms. We used the so-called *Jaccard index*  $J$ . For  $C, D \in V^n$  (where  $V$  is a set of objects)  $J(C, D)$  is computed like this:

$$J(C, D) = \frac{|C \cap D|}{|C \cup D|}$$

The intersection of vectors contains those elements that are in both vectors in the same position (that is with the same index), whereas the union contains the elements that are in  $C$  or  $D$ . Hence an  $i$ -th element of the vector  $C$  that is also the  $i$ -th element of  $D$  is an element of intersection. For instance, consider the vectors  $C = (1, 1, 3, X)$  and  $D = (1, 3, R, X)$ . The intersection contains the first and the last element of both vectors. Hence its cardinality is equal to 2. The union contains six elements. Thus the Jaccard index  $J(C, D) = \frac{2}{6} = 0,33$ .

#### 4.2.2. Methods of clustering

Having defined and computed the distance and similarity of our data, we can now cluster the data. To this end we applied the standard methods of  $k$ -means and  $k$ -medians.

*K-means* clustering is a method that aims to partition objects into  $k$  clusters in which each object belongs to the cluster with the closest means serving as a prototype representative of the cluster.

*K-medians* clustering is a variant of  $k$ -means method where instead of calculating the mean for each cluster to determine its centroid, one instead calculates the median with the most frequent value. This has the effect of minimizing error over all clusters with respect to the 1-norm distance metric, as opposed to the square of the 2-norm distance metric (which  $k$ -means does).

Hence both the methods consist of three steps:

1. The algorithm first chooses  $k$  objects as the means of clusters.
2. Each object is assigned to the closest mean
3. The new means are calculated.

The steps 2 and 3 are repeated until no more changes are made.

The proposed algorithm uses Lloyd-style iteration which alternates between an expectation (E) and maximization (M) step, making this an Expectation–maximization algorithm. In the E step, all objects are assigned to their nearest median. In the M step, the medians are recomputed by using the median in each single dimension.

The problem is computationally difficult, NP-hard. However, there are several efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, they both use cluster centers to model the data. However, *k*-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes. This relates to the *k*-median problem which is the problem of finding *k* centers such that the clusters formed by them are the most compact. Hence the distance of each object in the cluster to the mean object is minimal. The criterion function formulated in this way is sometimes a better criterion than that used in the *k*-means clustering algorithm, in which the sum of the squared distances is used.

*K*-means with iterated *splitting in half* is a variant in which at the first step all the input objects belong to one cluster which is split into two clusters. Using a given criterion (the size of cluster, its quality, etc.) we chose one of these clusters. The chosen cluster is again split into two halves, and we proceed in the same way till the *k* number of clusters is obtained.

In eLogika we also applied *hierarchic clustering*. This is an algorithm that applies strategies of two types: *agglomerative* and *divisive*. The former is a “bottom up” approach: each object starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. There are several variants of this method according to the way we merge the clusters. We can merge clusters with the closest centroid, or the clusters whose nearest (or most remote) objects have the least distance. Divisive strategy is a “top down” approach: all objects start in one cluster which is divided into two clusters that are again divided, and so on. The splits are performed recursively as one moves down the hierarchy. An example of such a method is also *k*-means with splitting in half.

As an input for these algorithms there are objects that we want to cluster and the number *k* of clusters to be created. After having created *k* clusters the algorithm halts. Another variant of the algorithm does not have a specified number *k* of clusters to be created, and the algorithm merges or divides the clusters as long as possible. As a result we obtain a hierarchical tree illustrating the way in which clusters were created. The root of the tree is the cluster containing all input objects, particular nodes are smaller clusters and the leaves are the objects. A user then decides which level is the best one to be used.

In order to cluster categorical data, the algorithm *ROCK* (see [9]) compares objects as for their similarity that can be computed for instance by Jaccard index. Input data for this algorithm consist of the objects to be clustered and a threshold similarity  $\theta$  that takes values in the interval  $(0, 1)$ . If two objects are more similar than the user specified threshold  $\theta$ , the algorithm creates a link between these objects. The clusters are then created in such a way as to minimize links between clusters so that most of the links are between the objects inside a cluster. The user specified threshold  $\theta$  is applied also as a criterion for the functions that determine which clusters should be merged together.



The *Rock* algorithm is said to be apt and effective for categorical data. Yet we voted for its variant *QuickROCK* which is a simplified and quicker version of the algorithm *ROCK*. *QuickROCK* also compares objects as for their similarity but it creates clusters in such a way that there are no links between clusters. Again, user defined threshold  $\theta$  determines which objects are linked together. This algorithm is much quicker and more suitable from the user point of view, because in this case there is a limited number of values that  $\theta$  can take. These are the values of Jaccard index for a given size off a vector. For instance, if we compare objects according to the values of their five attributes, then  $\theta$  can take only six values the two vectors can have in common: .

This makes user decision easier, though the results are a bit different from those of the *ROCK* algorithm. As an output the algorithm produces one or more clusters which depend on the user specified threshold  $\theta$ . If a too low  $\theta$  is specified then just one cluster is produced, because there are too many links between objects. On the other hand, a too high value of  $\theta$  (for instance  $\theta = 1$ ) causes that many clusters are produced, but all the objects in one cluster have identical values of attributes.

This way is suitable for pre-processing records in order to create associative rules. For instance instead of 100 identical records we obtain just one cluster that represents the records. In this way we do not lose anything and the resulting analyses are faster.

#### 4.2.3. Decision trees

Another tool that we applied is a decision-tree method serving both for prediction and analysis. It is a decision support tool that applies a tree-like graph of decisions and their expected consequences, for details see [3], [4], [6]. The algorithms of this tool generate hierarchical tree-like structures in which internal nodes represent tests on attributes, each branch represents outcome of test and each leaf node represents class label (decision taken after computing all attributes). A path from root to leaf represents classification rules. Hence each node divides records according to the value of a tested attribute. The tree is generated in a way that is an optimal division of records with respect to the chosen attribute.

One of the algorithms to compute the decision tree is ID3. This algorithm analyses a set of objects (vectors) with respect to a given attribute the values of which are to be predicted by the tree, and produces the least tree that satisfies the following conditions. The root of the tree contains all the input objects and each node divides the objects according to the values of one attribute so that the leaves contain objects with the same value of the predicted attribute.

The attribute the values of which divide the objects is determined by means of entropy  $S$  that is a measure of disorder. The entropy is calculated like this:

$$S = - \sum_i^n P_i \ln P_i$$

where  $P_i$  is the probability that the object in the analysed set has the value of the predicted attribute equal to  $i$ .

Decision trees are well arranged to illustrate the analysis and easy to interpret. The closer a node to the tree root is, the more important a given attribute is for the division of objects. Moreover, the shorter a given branch is the less attributes influence the predicted attribute.

#### 4.2.4. Association rules

Association rules (see [3]) are apt for instance for the prediction of the results of tests, because these rules are construed to discover hidden relationships among data. To this end we applied the *Apriori algorithm* that looks frequent combinations among data up. If we find such frequent combinations we can create the association rules using those matches that have sufficiently high number occurrences (support). Then we compute for these matches their support and confidence:

$$\begin{aligned} \text{support}(X \Rightarrow Y) &= P(X \cup Y) \\ \text{confidence}(X \Rightarrow Y) &= P(Y|X) = \frac{|X \cup Y|}{|X|} \end{aligned}$$

Here  $X$  and  $Y$  are values of attributes,  $P$  is probability. The support of a rule  $X \Rightarrow Y$  is determined by the percentage of records that contain both  $X$  and  $Y$ . The confidence of a rule  $X \Rightarrow Y$  is given by the conditional probability. It is the percentage of records  $X$  which contain  $Y$ .

*Example.* Suppose we have a database containing records of 500 students and their results in particular courses. By analysing these data we find out that 100 students obtained the grade A both from mathematics and logic (hence this match has a support 20%). Then we can make a hypothesis that if somebody has an excellent result in mathematics then they have an excellent result in logic as well, and vice versa, and to compute the confidence of this hypothesis. If the confidence is high, we conclude that there is a dependency between mathematical and logical skills. Or we can find out that though the support of this hypothesis is rather high, its confidence is low and the dependency is not sufficiently demonstrated.

The input for computing association rules are the objects (vector data) and the threshold value of support and confidence. The Apriori algorithm first looks the frequent matches up that is those matches that have a support higher than the given threshold. Then using the supported matches the rules are created. Finally those rules are voted for that have a confidence higher than the user-given threshold.

Hence by applying the method of association rules we can discover dependencies among data, like for instance the dependency between obtained credits in a given test and the time needed to work out the test, or between the results of those who executed several training tests and the others, etc.

Our tests are performed in different dates and times so that since we have quite a lot of data we can ask whether there are some dependencies between the results and the date and time of executing the test. We can also look for the dependencies between the number of students who executed a test, the form of study (distant vs. present) and the results.

Last but not least we ask whether with the growing number of attempts to succeed in the test the success rate is growing as well. Moreover, we ask whether those students who repeated the test twice or three times managed to obtain better results in the second and third attempts.

4.3. The results of the analysis of questions and answers

The results of questioning and answering were the first records that we analysed. We believed that this is the most important area to start with, because we wanted to evaluate the prepared database of questions and answers from which the tests are generated as for the difficulty of particular thematic groups.

We statistically analysed the results of tasks/questions and calculated an average number of correct answers. In order to obtain only those records that are reflective of the results we selected those questions that were used at least ten times. There were 132 such questions. Figure 3 illustrates average success rate of these questions.

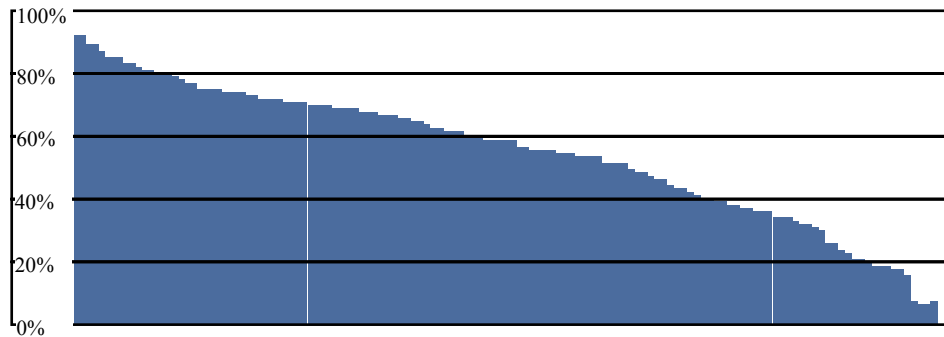


Figure 3. Average success rate of test questions

Looking at the graph we can see that though the distribution of particular tasks in the database is rather well balanced, there are a couple of difficult tasks with average success rate less than 10%. Hence we checked these tasks and in a few cases corrected their specification that was rather ambiguous and not clear. In some cases we also simplified the tasks.

Then we analysed the set of questions and students' answers by means of decision trees, which turned out to yield good and informative results. Each task consists of correct and incorrect answers to the given question. The students should mark the correct ones, leaving the incorrect unmarked. The simple decision tree contained just two attributes, *correct* answer and *incorrect* answer, and we tried to predict the success rate. Figure 4 depicts the obtained decision tree.

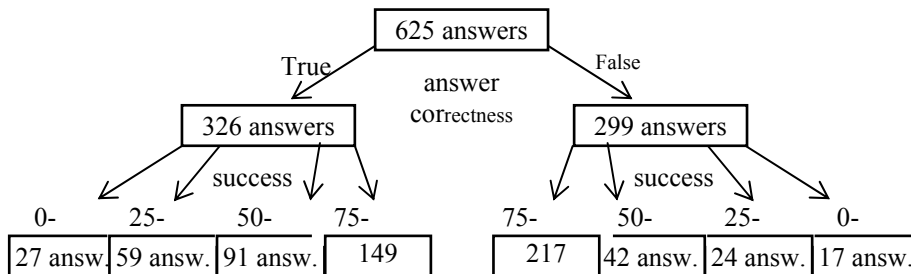


Figure 4. Decision tree illustrating the dependency between correct/incorrect answers and the success rate

We can extract from this figure, for instance, a piece of information that incorrect answers have much greater success rate although there are less incorrect answers than the correct ones. We used this result and adjusted the tasks so that incorrect answers were less obvious.

## 5. Conclusion

In this paper we introduced our e-learning system *eLogika* for Mathematical logic. The first prototype of the system has been used four years ago. The implementation of the system and its introduction to the teaching process was a necessary step, because otherwise the teachers could hardly manage the course due to the great amount of students enrolled in the courses on logic in our department. Yet this was not the only goal of the development of the system. We also wanted to make logic attractive for students. Thus since the introduction of *eLogika* into our education process we keep improving and tuning the system. In particular, we are currently including the elements that involve ‘*gamification*’ in order to promote cooperative work and competition among the students. In general, mathematical logic is not a favourite subject, although in our opinion it is a very attractive and in a way easy subject provided one understands its basic principles. Thus we wanted to motivate students for individual and creative work. For instance, we introduced financial rewards for the best students with best results, or for the most interesting solution of a group of tasks, or even for discovering interesting tasks, and so like. This was a success. The students began to spontaneously create groups that study together, look for good solutions of logical puzzles, and keep training logic. As a result, the anonymous evaluation of the course Mathematical logic in our information system became much more positive than before, and the success rate in final exams increased.

Yet this is still work in progress, and *eLogika* is a dynamic system that keeps being developed and tuned. In particular, we want to improve the tests by including more creative tasks with sample solutions, like using resolution method (or Hilbert calculus, natural deduction) find a direct proof of this or that argument. Currently there are such tasks in the database, but they are evaluated merely manually by a teacher. Of course, a manual checking of such tasks will always be necessary, but we want to navigate a student to a correct solution by providing sample proofs, or hints for the choice of axioms and rules. We also keep enriching *eLogika* database by new examples, exercises and tasks, we are looking for new and interesting logical tasks/puzzles, and we keep thinking of motivating elements that might increase students’ interest in logic.

**Acknowledgements.** This research has been supported by the European Social Fund and co-financed the state budget of the Czech Republic, projects No. CZ.1.07/2.2.00/28.0216 “Logika: systémový rámec oboru v ČR a koncepce logických propedeutik pro mezioborová studia (Logic: the Development of the Discipline and Basic Logic Courses)” and No. CZ.1.07/2.2.00/28.0209 “Elektronické opory a e-learning pro obory výpočtového a konstrukčního charakteru (“Computer-aided-teaching for computational and constructional exercises)”. We are grateful to the students who helped us to develop the eLogika system by implementing particular modules, whether within their diploma theses or as members of the developing team.

## References

- [1] M. Duží, M. Menšík, V. Hernas: E-Learning support for logic education. In *ICT for Competitiveness 2012*, Vymětal, D., Suchánek, P. (eds.), Karvina, 83-89.
- [2] M. Duží, M. Menšík, M. Číhalová, L. Dostálová: E-learning support for logic education. In *Digital Enterprise and Information Systems*, E. Ariwa and E. El-Quawasmeh (eds.), Berlin, Heidelberg: Springer-Verlag, CCIS vol. 194, (2011), 560-568.
- [3] J. Han and M. Kamber: *Data mining: concepts and techniques*. Morgan Kaufmann Publishers, San Francisco, 2006 (2<sup>nd</sup> ed.).
- [4] M. Berry: *Data mining techniques*. Oxford University, Indianapolis, 2004 (2<sup>nd</sup> ed.).
- [5] V. Rainardi: *Building a data warehouse with examples in SQL Server: processes, suitability and applications*. CA: Apress, Berkeley, 2008.
- [6] R. Bergmann: *Experience management: foundations, development methodology, and Internet-based applications*. Springer, New York, 2002.
- [7] A. Merceron, K. Yacef: Educational data mining: A case study, *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology*, IOS Press, (2005), 467-474.
- [8] C. Romero, S. Ventura: Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications* 33 (2007), 135-146.
- [9] S. Guha, R. Rastogi, K. Shim: ROCK: a robust clustering algorithm for categorical attributes. *Information Systems* 25, 5 (2000), 345-366.
- [10] *The OpenProof Project*. Stanford’s Center for the Study of Language and Information (CSLI). Retrievable at <http://ggwww.stanford.edu/NGUS/Openproof/>.

# A Data-driven Axes Creation Model for Correlation Measurement on Big Data Analytics

Takafumi NAKANISHI<sup>a,1</sup>

<sup>a</sup>*The Center for Global Communications (GLOCOM),  
International University of Japan, Japan*

**Abstract.** In this paper, we design a new model for Big data analytics – data-driven axes creation model. In Big data environment, the one of the important technologies is a correlation measurement. We cannot define a protocol of measurement on Big data era, because there are many varieties of data. However, almost current data analytics and data mining method cannot apply to Big data environment, because the big data environment is opened assumption and we have to consider new methods for opened assumption. That is, we have to design a new data-driven axes creation model for correlation measurement method. Our proposed model creates axes for correlation measurement on Big data analytics. Specifically, this model infers in the Bayesian network and measures correlation in the coordinate axes. Therefore, this model maps the Bayesian network into measure correlation mutually. This model contributes to a paradigm shift of Big data analytics.

**Keywords.** Data-driven processing, axes creation, correlation, Bayesian network, graphical model

## Introduction

Most of people are playing a major role in the dynamics of online systems. On the one side, we concentrate web application such as Twitter, YouTube, Facebook, etc. Sometimes these are called CGM. On the other side, there are a lot of fragmental data which are created by each person's device or which are created by amount of sophisticated sensors for science curiosities. Briefly speaking, we not only are retrieving but also creating these data every day. Mounts of various fragmental data are creating. We call “Big Data Era”. We doubt whether they became activation of the activity productive now that can distribute or search much data. In order to connect these resources to productive activity, we do not forget “discovery” not retrieval.

First, for leading the solutions, we have to marshal what Big Data is. We consider that the Big Data includes two prominent types of direction for ICT research purposes.

The one is the scale and speed issue of data processing. A lot of researchers have done this theme such as HPC, parallel distributed processing, and etc. Another is a schemaless data processing issue. It is important to real-timely discover the answers or clues for a user. A system has to create an appropriate schema from the data themselves

---

<sup>1</sup> Corresponding Author.

given by a user's query. Until now, data organized on the database schema. Currently, there are only fragmental various data on the web. This is a big paradigm shift. That is, it is necessary to create schema and data structures corresponding to user's required processing after a user inputs some queries. We have to shift the system from closed assumption to opened assumption. In the case of the closed assumption, the schema was designed in advance in consideration of orthogonal or independence. In the case of the opened assumption, we cannot care orthogonal and independence when the system dynamically creates schema.

The primitive of Big Data era is changing the systems from closed assumption to opened assumption. It is a limitation of expanding current system architectures. The disintegration of absolute sense of value is the most important. We lost an absolute hero, a cool rock star, a cute idol, etc. By the change of the role of the Web, it is important for us to create and spread each work and idea in each sense of values at a world. We will call a meme the work created on the basis of the individual sense of values, and an idea. We cannot evaluate good or bad absolutely. We have to evaluate by relatively comparing. Therefore, you have to stop strange index, schema, semantic web, Linked Open Data, etc., because they are only your individual sense of values, they do not contribute to other people. The other people evaluate their individual sense of values by relatively comparing. Their roles have finished. On this background, it is important to realize a new model for Big data environment.

In this paper, we design a new model for Big data analytics – data-driven axes creation model. In Big data environment, the one of the important technologies is a correlation measurement. We cannot define a protocol of measurement on Big data era, because there are many varieties of data. However, almost current data analytics and data mining method cannot apply to Big data environment, because the big data environment is opened assumption and we have to consider new methods for opened assumption. That is, we have to design a new data-driven axes creation model for correlation measurement method.

Our proposed model creates axes for correlation measurement on Big data analytics. Specifically, this model infers in the Bayesian network and measures correlation in the coordinate axes. Therefore, this model maps the Bayesian network into measure correlation mutually. This model contributes to a paradigm shift of Big data analytics.

This paper is organized as follows. In section 1, we represent the principle of Big data and Big data analytics. In section 2, we represent a data-driven axes creation model. This model estimates appropriate axes of data set by Bayesian estimation and map the data into coordinate axes consisting of estimated axes. Finally, we conclude in section 3.

## **1. Principle of Big Data and Big Data Analytics**

We have to focus on Big data analytics, which is completely different from such current data analytics as data mining technology. The key issues of Big data analytics are heterogeneity, continuity, and visualization.

### *1.1. Definition of Big Data*

Recently, not only business people, but also researchers are focusing on Big data, which is defined by three Vs [1][2]:

- volume: large amounts of data
- variety: different forms of data, including traditional database, images, documents, and complex records
- velocity: data content constantly changing through the absorption of complementary data collections and from streaming data from multiple sources

These 3V's definitions are on the viewpoint of infrastructures such as High Performance Computing and parallel distributed processing. These researches have finished. Because, big ICT companies such as Google, Amazon, Facebook, etc. operates these infrastructures as actual systems. What we have to consider is Big data infrastructures as a social problem. For example, Big data infrastructures need much electric power. The one of the important problems is how to operate Big data infrastructures without tragedy such as Fukushima. However, each big company is solving this problem. For example, Facebook will operate a Big data infrastructure by 100% wind-generated power as a system by 2016 [3]. It is impossible in Japan at least whose liberalization of electric power is not enough.

However, some people use keyword "Big data". We computer scientist should consider this to be needs. What are these needs? The keyword "Small data"[4] tells us a hint.

The definition of Small data is described by [4] as follows: Small data connect people with timely, meaningful insights (derived from big data and/or "local" sources), organized and packaged – often visually – to be accessible, understandable, and actionable for everyday tasks. This definition is similar to the advantages and sales talks of Big data. We consider that these needs mean how to analyze non-schema data. In this context, the volume including the 3V is not related, it may be large, or may be small, or whichever may be sufficient.

In order to analyze appropriately such above description, a system has to correctly map into cyber world from real world. In the next section, I show the mapping from real world to cyber world.

### *1.2. Mapping from Real World to Cyber World*

The one of the important elements for Big data analytics, including small data case such as [4] is mapped correctly in cyber world from the real world. Figure 1 shows the relationship between real world and cyber world. Sensors aggregate real world situation as discrete data. However, real world is continuous. In order to correctly analyze in cyber world, a system has to be analyzed by using continuous value. Therefore, fitting or interoperating is very important.

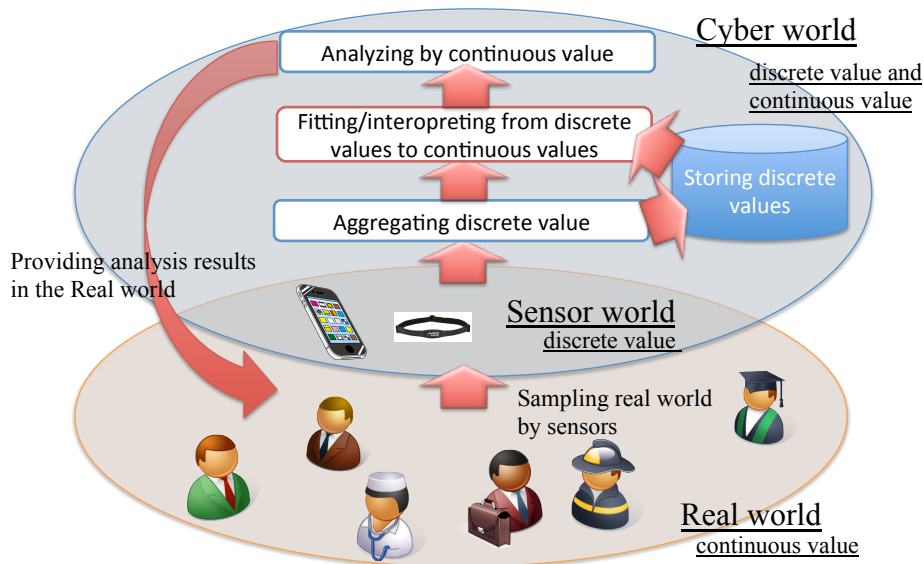
It is a very easy thing. For example, for listening to music, CDs and a CD player can be used. CDs have discrete sound data from the real world. Their music cannot be recognized without digital/analogue conversion by a CD player. This example shows us one important feature of Big data. Each piece only represents an instance of a certain state. Of course, higher sampling rates produce more correct data. However, the real



world is a continuous place. Unless the data are continuous, many things cannot be discovered, like the example of a CD's music.

Other issues include which axis to interpolate. In the example of a CD's, music, we can interpolate the time axis by a digital/analogue converter. Depending on the information provided by the data, we do not know which interpolation uses place information or which uses the temperature of air/water information.

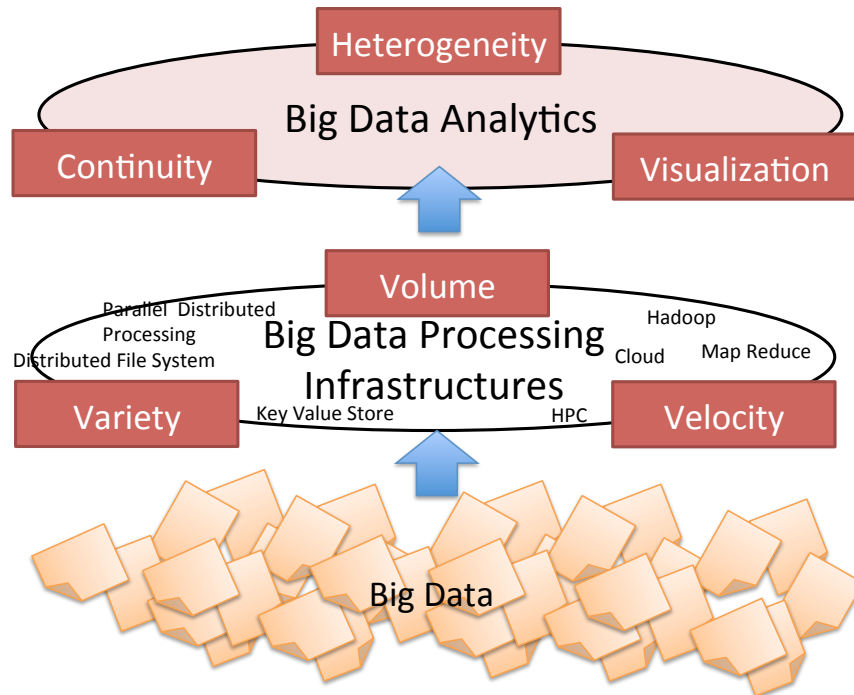
Finally, even though computer science researchers have researched approximation method from discrete to continuous values, they have never researched the selection or the creation of appropriate axes for continuous discovery. This is a critical theme of Big data analytics.



**Figure 1.** Relationship between real world and cyber world. Sensors aggregate real world situation as discrete data. However real world is continuous. In order to correctly analyze in cyber world, a system has to be analyzed by using continuous value.

On the other aspects, we have very variety of sensors on the real world because of spreading mobile terminals and accurate low price sensor. Such sensors will continue increasing in number from now on also. Can we design a schema such as situation? It is impossible. Therefore, a system has to various data on non-schema. In addition, the system has to integrate these heterogeneous data.

Finally, most sensor data are fragmental. A web page is one of the data in the cyber world. We enjoyed these data as contents. Such data are becoming the past things. Current data are more fragments. For example, each tweet on twitter consists of short sentences or words; sensor data consists of a numerical value; etc. This means we cannot enjoy one data as contents. Therefore, we have to shift the paradigm of search engine. Currently, most search engines provide a list of data corresponding to the user's keywords by pattern matching. However, a user is beginning to lose user's interest by each data as contents. Almost users want to see the overview or trend of these data in the user's interest. It is not enough to see the overview or a trend of focused data by representation of a list.



**Figure 2.** Relationship among three Vs of Big data definition and Big data analytics definition: heterogeneity, continuity, and visualization.

We consider that it is important to create contents by using fragment data in cyber world. It means that visualization is changing. By user's query, a system aggregate necessary data, analysis these data and create visualization along with analysis. Therefore, a system creates new contents for user corresponding to user's query automatically. In the near future, a system creates actual something in real world by 3D printers and other methods. This is mutual mapping between real world and cyber world.

### *1.3. Three Primitive of Big Data Analytics*

It is important to discover answers or clues for users in real time. A system has to create an appropriate schema from the data themselves given by user queries. Until now, data have been organized based on database schema. Currently, only various fragmentary data exist on the web.

This is a huge paradigm shift. We must create the schema and data structures that correspond to the processing required by users after they input queries. We have to shift the system from designing closed assumptions to opened assumptions.

Heterogeneity, continuity, and visualization are the most critical features of Big data analytics, which provides a scale and connection merits based on them. No current data analysis methods are based on opened assumptions. Big data analytics provide a new data analysis method based on opening assumptions. Below, we discuss the inconsistencies caused by continuing to use the current methods. Figure 2 shows the

relationship between the three elements (volume, variety, velocity) of Big data's definition and its analysis features (heterogeneity, continuity, visualization).

### 1.3.1. Heterogeneity: Big Data Analytics Features

In Big data analytics, heterogeneity is different from the type to which the Big data definition belongs. The variety of Big data definitions includes such content as images, sounds, documents, etc. Its heterogeneity includes such data fields as news, entertainment, technology, and science, all of which are semantic aspects.

In Big data analysis, reasonable correlations must be discovered between heterogeneous fields. Currently, semantic web technologies [5][6][7] or association rule extraction technologies [8] are generally used. However, in Big data analytics, there are three inconsistencies because the Big data environment is an opened assumption not a closed assumption [9].

Until now, computer science researchers have based their ideas on closed assumptions by freely linking and interconnecting each object. For example, the interconnection between element  $a_i$  in set  $A$  and element  $b_j$  in set  $B$  for  $A \cap B \neq \emptyset$  remains a closed assumption. However, users, especially data intensive scientists, do not require such knowledge. They have to consider new discovery methods in opened assumptions, where  $A \cap B = \emptyset$ .

Note that such inconsistencies only occur when extending the current methods introduced. We call these inconsistencies the Three Opened Assumption's Inconsistencies:

- (1) A relation does not guarantee the future.

For example, we can identify relationships among each set through data mining technology. Note that the results only represent the relationships of the present data. These relationships are not guaranteed if the system adds new records (data). Occasionally, researchers and users anticipate an uncertain value of a new record using extracted relationships. However, such usage is incorrect. Due to the insignificance of predicting uncertain values by data mining, we assume that sets  $A$  and  $B$  are attributes in the relational database and that  $a_i$  and  $b_i$  are the attribute values of each set. The data mining result is guaranteed if no updates occur. However, most tables undergo many updates. We assume  $k$  records in the database and that the numbers of each attribute value are  $k$ . The system performs data mining and extracts  $b_i = f(a_i)$ . This relation  $f$  is only guaranteed when there are  $k$  records in the database. If the number of records is  $k + 1$ , relation  $f$  is not guaranteed. Indexing relations, which are extracted by various methods, is meaningless for predicting uncertain or missing values.

- (2) A transitive relations is not true when a user connects links for heterogeneous fields.

With closed assumptions, we create or extract relationships in a set. In this case, the transitive and order relations are true. However, they are not true when we create or extract relationships over the sets. This phenomenon occurs when we use bridge ontology, semantic webs, linked data, etc. Each ontology in specific fields is unconsciously created in closed assumptions. These techniques connect ontologies in specific fields that are changed from closed to opened assumptions. Therefore, it may become possible to use these ontologies only by

connecting each element of each set. On the contrary, is the determinant possible when trying to create bridge ontology? Such determination is difficult.

- (3) No relations in heterogeneous fields can be discovered in set theory.

Should such relationships be indexed or aggregated? We might semantically discover new relationships, but how to discover them is not understood. Even if part of the relations of each element of the sets is known, the relations of all of the sets are not guaranteed. Moreover, the relation is not guaranteed when a new record is entered, even if the relation of the sets was previously guaranteed. Therefore, even if we can retrieve the relationships, their discovery is impossible by inference and reasoning because only the relationships that we have discovered are effective; transitive and order relationships are not guaranteed. The transitive and order relations are not true when we create or extract relationships over the sets. This result is disappointing. However, it represents a paradigm shift from closed to opened assumption systems.

By connecting each element of each set, it may become impossible to use these ontologies. Computing some systems is very inefficient with bridge ontology and linked data. On the contrary, is the determinant in the case of trying possible for the author of bridge ontology? Determining this is very difficult. Of course, schema mapping has the same problem, since RDB has a relation. Discovering new relations by inference and reasoning is difficult.

### *1.3.2. Continuity: Features of Big Data Analytics*

Most big data come from sensors. For social media, each human action is part of the Big data from each human sensor. It is important to aggregate every second such massive and various sensor data in the Big data processing infrastructures. However, there is a restriction on the sampling rate. For more realistic analysis, we must approximate the aggregated discrete data to continuous value data, because the real world is continuous.

For example, for listening to music, CDs and a CD player can be used. CDs have discrete sound data from the real world. Their music cannot be recognized without digital/analogue conversion by a CD player. This example shows us one important feature of Big data. Each piece only represents an instance of a certain state. Of course, higher sampling rates produce more correct data. However, the real world is a continuous place. Unless the data are continuous, many things cannot be discovered, like the example of a CD's music.

Other issues include which axis to interpolate. In the example of a CD's music, we can interpolate the time axis by a digital/analogue converter. Depending on the information provided by the data, we do not know which interpolation uses place information or which uses the temperature of air/water information.

Finally, even though computer science researchers have researched approximation method from discrete to continuous values, they have never researched the selection or the creation of appropriate axes for continuous discovery. This is a critical theme of Big data analytics.

### *1.3.3. Visualization: Big Data Analytics Features*

Visualization is important for Big data analytics for many different reasons. For example, a Google search provides a list that represents an appropriate data set. Is such a list of representations satisfactory? In Big data environments, not every piece of data can be clicked on and checked because we are not interested in such details. Since we want to identify the trends of the whole data set, visualization is a crucial issue for Big data analytics. We can use various visualization methods created by researchers.

However, another issue remains. What kinds of visualization provide better correspondence to user queries? A system has to select, create, and provide appropriate visualization for users. To choose visualization, a system has to know which axis to focus on. For example, a graph that makes X-axis times is good for seeing the changes of a value of time. Therefore, realizing the creation or a selection method of axes that correspond to user queries is important. When we realize systems with such methods, they can select the appropriate visualization. On Big data analytics, visualization changes are based on user purposes.

### *1.3.4. Common Issue for Three Primitive of Big Data Analytics*

We have to consider how to construct Big data analytics for heterogeneity, continuity, and visualization features. One critical issue is from what viewpoint to see Big data. We have no axes that measure weather data are appropriate. In Big data environments, systems have to provide not only appropriate data set but also axes as viewpoints that solve user queries. How do the system extract axes?

In the next section, we propose a data-driven axes creation model for correlation measurement.

## **2. A Data-driven Axes Creation Model**

We propose a new axes creation model in this section. This model estimates appropriate axes of data set by Bayesian estimation and map the data into coordinate axes consisting of estimated axes.

In section 2.1, I introduce a basic graphical model [10] and formulations. In section 2.2, I propose a data-driven axes creation model, especially, I describe mapping between graphical model in Bayesian estimation and correlation measurement on coordinate axes. Finally, in section 2.3, I show a new data analytics flow for Big data era.

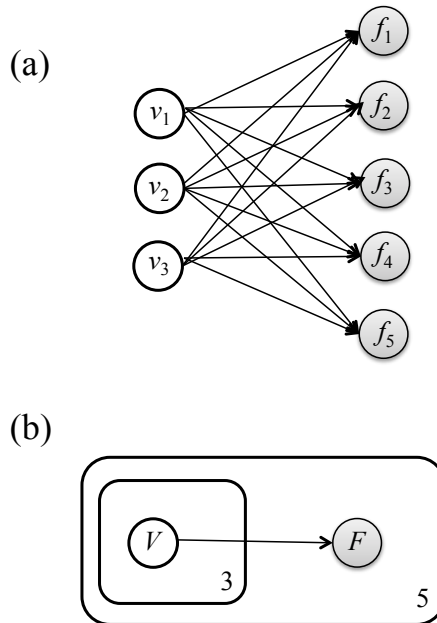
### *2.1. Graphical Model*

#### *2.1.1. The Overview of Graphical Model*

We introduce the brief overview of graphical model. Figure 3 shows the basic model represented in graphical models. A graphical model is a probabilistic model for which a graph denotes the conditional dependence structure between random variables. In this paper, we describe the case of Bayesian network model. The Bayesian network is represented in the directed acyclic graph (DAG). The Bayesian network can easily derive formula which we have to solve.

The colored nodes  $f_i$  in Figure 3 are the observable value. It means we can obtain some values of the colored nodes. In contrast, the nodes  $v_j$  call latent variable or hidden variable. The squares call plates which represent in condensed formula. (a) and (b) in Figure 3 are the same in meaning when  $V=\{v_1, v_2, v_3\}$  and  $F=\{f_1, f_2, f_3, f_4, f_5\}$ .

Generally, each probability is represented in  $p(v_j)$ ,  $p(f_i)$ ,  $p(V)$ , and  $p(F)$ . The reason why the graphical model is effective is that we can represent joint probability in a following formula:



**Figure 3.** A simple example of a graphical model.

$$p(V, F) = p(V|F)p(F).$$

Therefore,  $p(v_j)$  can be derived by following a following formula:

$$p(v_j) = \sum_{i=1}^5 p(v_j|f_i)p(f_i).$$

For example,  $F$  is a set of features represented in data, contents or something.  $V$  is a set of events which can be detected by feature set  $F$ . We would like to derive some weight of each edge. That is, we would like to estimate the value  $p(V|F)$ . When we would like to estimate  $p(V|F)$ , we usually use maximum likelihood estimation method[11], variational Bayesian estimation[10], Markov Chain Monte Carlo method (MCMC)[12], etc. We can estimate each  $p(v_j)$  from input each  $p(f_i)$ .

This is the one of general methods in machine learning. For example, the technique is used in sound recognition, image recognition, topic extraction, etc. Especially, it is the Latent Dirichlet allocation (LDA)[13] which is the one of the topic model, when  $V$  is multinomial distribution, we add one more node which is the topic distribution and we apply Figure 3 to hyper parameters.

Anyway, we have a lot of method of estimation for  $p(V|F)$ . It is a big contribution for a data-driven axes creation model.

### 2.1.2. Variational Bayesian Estimation

In this section, we show the one of the estimation methods [10], variational Bayesian estimation which is used in this paper. Please note that our method can be applied to other estimation method.

It expresses with the stochastic variables  $X$  and  $Z$ . In addition,  $X$  is known stochastic variables and  $Z$  is unknown variables. The unknown variable  $Z$  denotes marginalization as follows.

$$\begin{aligned} p(X) &= \int_Z p(X, Z) dZ \\ \log p(X) &= \int_Z \log p(X, Z) dZ = L(q) + KL(q||p) \geq L(q) \\ L(q) &= \int_Z q(Z) \frac{p(X, Z)}{q(Z)} dZ \\ KL(q||p) &= - \int_Z q(Z) \frac{p(Z|X)}{q(Z)} dZ \end{aligned}$$

$KL(q||p)$  is a Kullback–Leibler divergence. Therefore, The Kullback–Leibler divergence is a minimum value when  $q(Z)=p(Z|X)$ . However, it is hard to solve  $p(Z|X)$  distribution.

Here, we apply to the mean field approximation. The mean field approximation are represented as follows when a set of unknown variable  $Z=\{z_1, z_2, \dots, z_k\}$ :

$$q'(Z) = \prod_{i=1}^k q_i(z_i)$$

The  $q'(Z)$  can be represented by the Kullback–Leibler divergence. The approximate solution which should be calculated is equivalent to the minimum of the following formula:

$$KL(q'||q) = \int q'(Z) \log \frac{q'(Z)}{q(Z)} dZ$$

The  $q'(Z)$  is substituted for  $L(q)$ :

$$\begin{aligned} L(q) &= \int_Z \prod_{i=1}^k q_i(z_i) \frac{p(X, Z)}{\prod_{i=1}^k q_i(z_i)} dZ \\ &= \int_Z q_j(z_j) \left\{ \int \log p(X, Z) \prod_{i \neq j} q_i(z_i) dz_i \right\} dz_j \\ &\quad - \int_Z q_j(z_j) \log q_j(z_j) dz_j + const \\ &= \int_Z q_j(z_j) \log \frac{\tilde{p}(X, z_j)}{q_j(z_j)} dz_j + const = KL(q_j||\tilde{p}) + const \end{aligned}$$

Maximization of  $L(q)$  is equivalent to minimization of Kullback–Leibler divergence  $KL(q_j||\tilde{p})$ . The optimal solution  $q_j^*(Z_j)$  is calculated as follows:

$$\log q_j^*(Z_j) = \int \log p(X, Z) \prod_{i \neq j} q_i(Z_i) dZ_i + const = \mathbb{E}_{i \neq j} [\log p(X, Z)] + const$$

$$q_j^*(Z_j) = \frac{\exp(\mathbb{E}_{i \neq j}[\log p(X, Z)])}{\int \exp(\mathbb{E}_{i \neq j}[\log p(X, Z)]) dZ_j}$$

## 2.2. Mapping between Graphical Model in Bayesian Estimation and Correlation Measurement on Coordinate axes

### 2.2.1. Basic Theorem of Our Model

We look back to Figure 3. There are two set  $V = \{v_j\}$  and  $F = \{f_i\}$ . Each element between  $\{v_j\}$  and  $\{f_i\}$  which is represented in nodes is connected by edges. The each value of each edge are represented in  $p(v_j|f_i)$ .

In conclusion, a conditional probability set of  $p(v_j|f_i)$  is a mapping operator from  $F$  to  $V$  on the viewpoint of linear algebra. Therefore, estimation methods in machine learning such as variational Bayesian estimation [10], etc. automatically create axes for correlation measurement as same as automatically create mapping operator. We show a basis for conclusions as follows. For generalization, the number of elements (nodes) of  $F$  is  $M$ ; the number of elements (nodes) of  $V$  is  $N$ . Figure 4 shows the graphical model represented in this case.

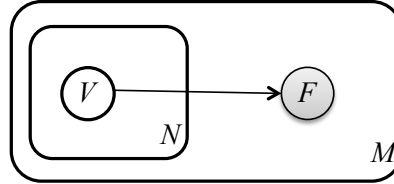


Figure 4. A graphical model for formulation.

The set of  $p(f_i)$  is represented as a vector  $\mathbf{x}$  which has  $M$  elements.

$$\mathbf{x} = (p(f_1), p(f_2), \dots, p(f_M))^T$$

The set of  $p(v_j|f_i)$  is represented as a  $M \times N$  matrix  $A$ .

$$A = \begin{bmatrix} p(v_1|f_1) & \cdots & p(v_1|f_M) \\ \vdots & \ddots & \vdots \\ p(v_N|f_1) & \cdots & p(v_N|f_M) \end{bmatrix}$$

The set of  $p(v_j)$  is represented as a vector  $\mathbf{y}$  which has  $N$  elements.

$$\mathbf{y} = (p(v_1), p(v_2), \dots, p(v_N))^T$$

We can derive relationship among  $\mathbf{x}$ ,  $\mathbf{y}$  and  $A$  as follows:

$$\mathbf{y} = A\mathbf{x}$$

It is because the following formulas are materialized.

$$p(v_j) = \sum_{i=1}^M p(v_j|f_i)p(f_i)$$

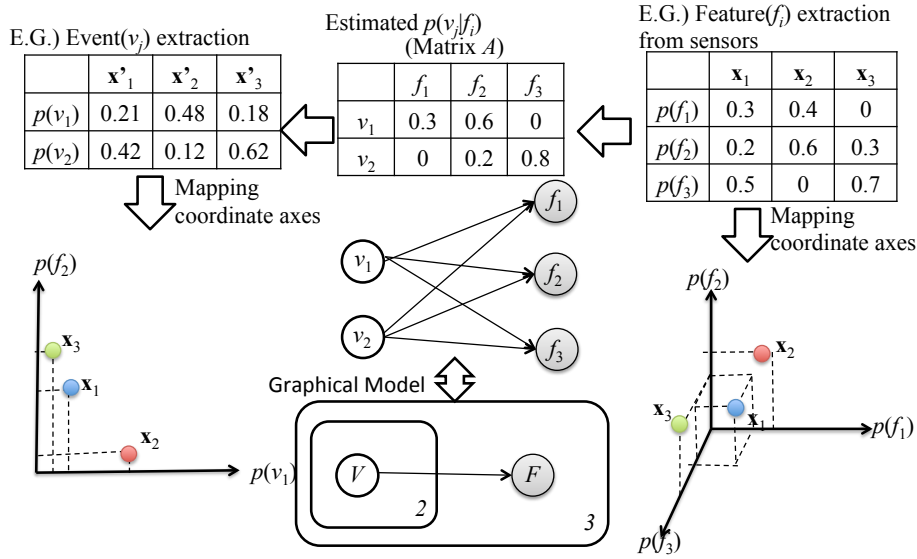
Therefore, a conditional probability set of  $p(v_j|f_i)$  is a mapping operator from  $F$  to  $V$  on the viewpoint of linear algebra. The estimation of  $p(V|F)$  is same as creating axes for correlation measurement and creating a mapping operator from  $F$  to  $V$ .

You may worry about independency of each element of  $V$ . When we use variational Bayesian estimation [10] for estimation, independency of each element of  $V$  is guaranteed. The variational Bayesian estimation uses the mean field approximation. The mean field approximation assumes that all unknown variable is independent.



### 2.2.2. Simple Example

In this section, we show a simple example of our model as shown in section 2.2.1. Figure 5 shows the simple example. This example is the case of mapping from features extracted by sensor to events. In other words, a system detects events from sensor data and measure correlation among past events and current event. By correlation measurement between past events and current detected event, we can predict future provisions. In this example, vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are past data set of each sensors  $f_i$ . These vectors have links to experiences such as actual events, provisions, etc. In the feature layer, we cannot similarity. We have to map from a feature space to an event space. As shown in section 2.2.1, The mapping operator consists of each  $p(v_j|f_i)$  value. We have to estimate these values. In this example, it assumes that we have estimated each  $p(v_j|f_i)$  value by using the variational Bayesian estimation and have created Matrix  $A$ . In this situation, a system obtains the data  $\mathbf{x}_3$ . By using this data, we would like to predicate in the future by using past events. In order to predicate, we have to map past data sets  $\mathbf{x}_1$  and  $\mathbf{x}_2$  and obtained data  $\mathbf{x}_3$  by using the mapping operator  $A$ ; we measure correlation among  $\mathbf{x}_1$  and  $\mathbf{x}_2$  and obtained data  $\mathbf{x}_3$ . When we can find similar data, we can predicate in the future by following links.



**Figure 5.** A simple example.

$\mathbf{x}_1$  and  $\mathbf{x}_2$  are past data sets of sensor  $f_i$ .  $\mathbf{x}_3$  is an observed current data. Here, we would like to predict based on past data sets  $\mathbf{x}_1$  and  $\mathbf{x}_2$ .  $\mathbf{x}_1$  and  $\mathbf{x}_2$  have links of actual events and provisions. When we have estimated each  $p(v_j|f_i)$  value, we can map each  $\mathbf{x}_k$  value into event ( $v_j$ ) space. We can predicate from past experiences when we can get similar events to  $\mathbf{x}_3$  situation in the event space. In the example case, we can see that  $\mathbf{x}_3$  is similar to  $\mathbf{x}_1$  in the event space. The  $\mathbf{x}_1$  has links about actual events and provisions. By this experience data, we can predicate in the future.

A vector  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\mathbf{x}_3$  is represented as follows:

$$\mathbf{x}_k = (p(f_1), p(f_2), p(f_3))^T$$

$$\mathbf{x}_1 = (0.3, 0.2, 0.5)^T$$

$$\mathbf{x}_2 = (0.4, 0.6, 0)^T$$

$$\mathbf{x}_3 = (0, 0.3, 0.7)^T$$

We have to estimate each  $p(v_i|f_j)$ . Generally, This boils down to the question of estimation of following formula by the variational Bayesian estimation.

$$p(V, F) = p(V|F)p(F)$$

The reference [13] shows efficient estimation model which calls Latent Dirichlet allocation (LDA). This model applies two hyper parameters  $\alpha$  and  $\beta$ . This model also adds axis distribution variance  $\theta$ . The axes  $p(\theta, V|F, \alpha, \beta)$  represents as follows:

$$p(\theta, V|F, \alpha, \beta) = \frac{p(\theta, V, F|\alpha, \beta)}{p(F|\alpha, \beta)}$$

$$p(\theta, V, F|\alpha, \beta) = p(\theta|\alpha)p(V|\theta)p(F|V, \beta)$$

That is, we estimate each parameter and each variance by the variational Bayesian estimation. In this example, it assumes that we have estimated  $p(\theta, V|F, \alpha, \beta)$  or  $p(V|F)$ . By Figure 4, the mapping operator  $A$  represents as follows:

$$A = \begin{bmatrix} p(v_1|f_1) & p(v_1|f_2) & p(v_1|f_3) \\ p(v_2|f_1) & p(v_2|f_2) & p(v_2|f_3) \end{bmatrix} = \begin{bmatrix} 0.3 & 0.6 & 0 \\ 0 & 0.2 & 0.8 \end{bmatrix}$$

We map past data sets  $\mathbf{x}_1$  and  $\mathbf{x}_2$  and obtained data  $\mathbf{x}_3$  into event space ( $\mathbf{x}'_1$ ,  $\mathbf{x}'_2$  and  $\mathbf{x}'_3$ ) by using the mapping operator  $A$ .

$$\mathbf{x}'_k = A\mathbf{x}_k = (p(v_{k1}), \dots, p(v_{kN}))^T$$

$$\mathbf{x}'_1 = A\mathbf{x}_1 = (0.21, 0.42)^T$$

$$\mathbf{x}'_2 = A\mathbf{x}_2 = (0.48, 0.12)^T$$

$$\mathbf{x}'_3 = A\mathbf{x}_3 = (0.18, 0.62)^T$$

In order to measure correlation among  $\mathbf{x}'_k$ , we can define some measurement definition. For example, we define inner product, distance and norm as follows:

$$innerProduct(\mathbf{x}'_k, \mathbf{x}'_{k'}) = \frac{\sum_{j=1}^N p(v_{kj}) p(v_{k'j})}{\sqrt{\sum_{j=1}^N p(v_{kj})^2} \sqrt{\sum_{j=1}^N p(v_{k'j})^2}}$$

$$distance(\mathbf{x}'_k, \mathbf{x}'_{k'}) = \sqrt{\sum_{j=1}^N (p(v_{kj}) - p(v_{k'j}))^2}$$

$$norm(\mathbf{x}'_k) = \sqrt{\sum_{j=1}^N p(v_{kj})^2}$$

This is very general and natural definition. You can define the other measurements. In this paper, we use an inner product as correlation measurement. We calculate inner product between  $\mathbf{x}'_3$  and others.

$$innerProduct(\mathbf{x}'_3, \mathbf{x}'_1) = 0.37275$$

$$innerProduct(\mathbf{x}'_3, \mathbf{x}'_2) = 0.201$$

In these results, current situation  $\mathbf{x}_3$  is similar to  $\mathbf{x}_1$ . We can predict in the future by following the links of the  $\mathbf{x}_1$ .

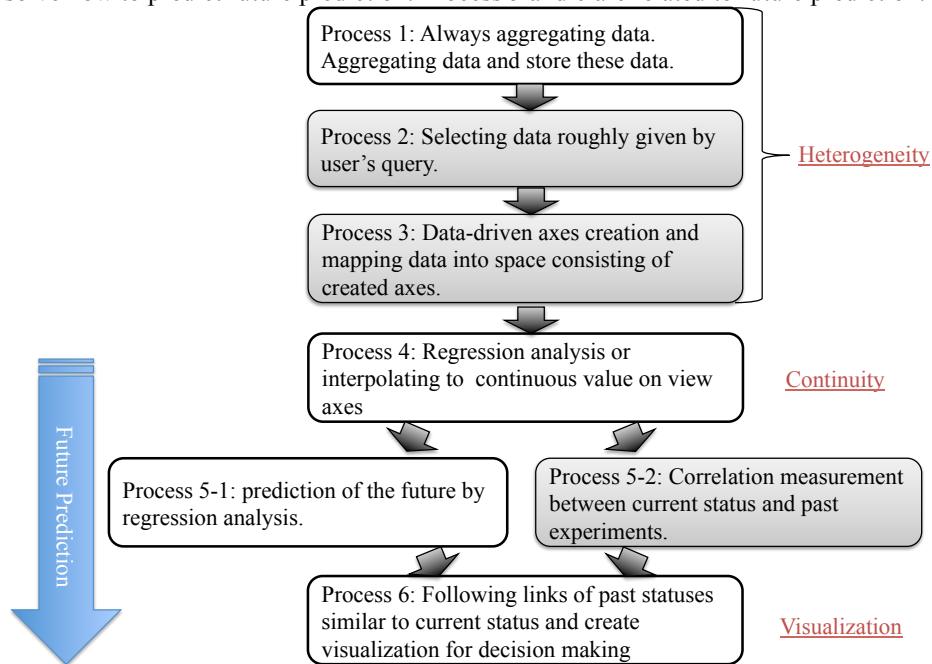
We showed a simple example. In this result, the estimation of conditional probability which is studied in the machine learning community, such as the variational Bayesian estimation is same as the creation of appropriate axes for correlation measurement and its mapping operator. In addition, we can define various measurements for correlation measurement. Therefore, we can realize a data-driven axes creation and its mapping operator creation.

In this method, we can use not only data-driven appropriate space creation, but also the interconnection between local knowledge base and global knowledge base. For

example, we can use interconnection between music data set and image data set by their impression and cross-cultural system. When we interconnect cross-domain or heterogeneous data, there are three inconsistencies because the Big data environment is an opened assumption not a closed assumption as shown in the reference [9]. In our model, a system automatically creates mapping operator and map the data in common measurement space. It does not occur the three inconsistencies. This is approximate method from opened assumption to closed assumption dynamically. This model contributes a paradigm shift of Big data analytics.

### 2.3. Data Analytics Flow for Big Data Era

Figure 6 shows a system process of Big data analytics. Colored node means related to this paper. The differences from old analytical methods are heterogeneity, continuity, and visualization. In addition, in order to satisfy user's needs of analytics, we have to solve how to predict future prediction. Process 5 and 6 are related to future prediction.



**Figure 6.** The overview of flow for Big data analytics.  
The colored nodes are related to this paper.

- Process 1: Always aggregating data  
Big data environment satisfies 3V (Volume, Variety, Velocity). Therefore, a system should aggregate various data every time.
- Process 2: Selecting data roughly given in the user's query  
A system roughly selects data given in the user's query. It is enough to be rough, because the selected data are used by creation of axes. It is enough to select data completely after deciding axes.

- Process 3: Data-driven axes creation and mapping data into space consisting of creating axes  
A system creates axes for correlation measurement of selected data. After that, a system map appropriate data into space consisting of creating axes.
- Process 4: Regression analysis or interpolating to continuous value on focused axes  
The aggregated data are discrete-valued data. In order to recognize real situation, we have to approximate continuous value of the aggregated data. We can get a latent context without discrete-valued data. In addition, we can get predicating values by regression analysis.
- Process 5-1: Prediction of the future by regression analysis  
In order to predict the values in the future, by regression analysis, a system approximates discrete-valued data as a function. Therefore, we can predict the future values.
- Process 5-2: Correlation measurement between current status and past experiments  
Another method is correlation measurement. A system measures correlation among past statuses and current status. By this process, we can see similar situation in past experiments. By this result, a system follows links of similar past situation. We can predict next events and provisions.
- Process 6: Following links of past statuses similar to current status and create a visualization for decision making  
It is important to visualize for representation in decision-making. A system provides not only current situation analysis, but also a prediction by process 5-1 or 5-2.

### 3. Conclusion

In this paper, we designed a new model for Big data analytics – data-driven axes creation model. In Big data environment, the one of the important technologies is a correlation measurement. We cannot define a protocol of measurement on Big data era, because there are many varieties of data. However, almost current data analytics and data mining method cannot apply to Big data environment, because the big data environment is opened assumption and we have to consider new methods for opened assumption. That is, we have to design a new data-driven axes creation model for correlation measurement method.

We also discussed about the principle of Big data and Big data analytics. The big data area is a huge paradigm shift. We must create the schema and data structures that correspond to the processing required by users after they input queries. We have to shift the system from designing closed assumptions to opened assumptions.

Heterogeneity, continuity, and visualization are the most critical features of Big data analytics, which provides a scale and connection merits based on them. No current data analysis methods are based on opened assumptions. Big data analytics provide a new data analysis method based on opening assumptions.

A future subject is establishing the new data analysis technique in Big data era along with three primitives – heterogeneity, continuity, and visualization.

## References

- [1] J. Berman, "Principles of Big Data Preparing, Sharing, and Analyzing Complex Information", Elsevier / Morgan Kaufmann, 2013.
- [2] M. Minelli, M. Chambers, A. Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
- [3] K. Finley, "Facebook Says Its New Data Center Will Run Entirely on Wind," WIRED, Nov.13,2013 <http://www.wired.com/wiredenterprise/2013/11/facebook-iowa-wind/>.
- [4] P. Greenberg, "10 Reasons 2014 will be the Year of Small Data", ZDNet, Dec. 2, 2013. <http://www.zdnet.com/10-reasons-2014-will-be-the-year-of-small-data-7000023667/>
- [5] T. Berners-Lee, Linked Data, W3C Design Issues, July 2006. From <http://www.w3.org/DesignIssues/LinkedData.html>; retr. 2012/11/26
- [6] C. Bizer, T. Heath, T. Berners-Lee, Linked Data – The Story So Far, International Journal on Semantic Web and Information Systems (IJSWIS), 5(3), 1-22, 2009.
- [7] M. Greaves, P. Mika, Semantic Web and Web 2.0, Journal of Web Semantics 6 (1), 1-3, 2008.
- [8] E. Gonzales, T. Nakanishi, K. Zettsu, Large-Scale Association Rule Discovery from Heterogeneous Databases with Missing Values using Genetic Network Programming, In Proc. of the 1st International Conference on Advances in Information Mining and Management, 113-120, 2011.
- [9] T. Nakanishi, K. Uchimoto, Y. Kiadawara, Inconsistencies of Connection for Heterogeneity and a New Relation Discovery Method that Solved them, In Proceedings of 12th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2013), pp. 521-528, 2013.
- [10] M. Christopher Bishop, Pattern Recognition and Machine Learning. Springer, 2006.
- [11] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society, Series B 39 (1): 1–38, 1977.
- [12] C. Andrieu, N. De Freitas, A. Doucet, M. I. Jordan, An Introduction to MCMC for Machine Learning, Machine Learning, 50, 5–43, 2003.
- [13] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet allocation, In Lafferty, John. Journal of Machine Learning Research 3 (4–5): 993–1022, 2003. doi:10.1162/jmlr.2003.3.4-5.993.

# An Adaptive Search Path Traverse for Large-scale Video Frame Retrieval

Diep Thi-Ngoc NGUYEN<sup>a,1</sup> and Yasushi KIYOKI<sup>a</sup>

<sup>a</sup> *Graduate School of Media and Governance, Keio University, JAPAN*

**Abstract.** Multimedia retrieval task is faced with increasingly large datasets and variously changing preferences of users in every query. We realize that the high dimensional representation of physical data which previously challenges search algorithms now brings chances to cope with dynamic contexts. In this paper, we introduce a fast search algorithm using utilization of inverted indexes for high dimensional metadata and build a large-scale video frame retrieval environment handling users' dynamic contexts of querying by imagination, and controlling response time. The search algorithm quickly finds an initial candidate, which has highest-match possibility, and then iteratively traverses along feature indexes to find other neighbor candidates until the input time bound is elapsed. The experimental studies based on the video frame retrieval system show the feasibility and effectiveness of our proposed search algorithm that can return results in a fraction of a second with a high success rate and small deviation to the expected ones. Moreover, its potential is clear that it can scale to large dataset while preserving its search performance.

**Keywords.** Large-scale multimedia retrieval, Video navigation, Inverted index, Local search, Adaptive response time

## Introduction

The design of our adaptive searching method aims to solve three issues in video data retrieval systems: large size of datasets, dynamic users' contexts in queries, and requirement of controlling response time. In this paper, we propose an improved inverted index technique to index metadata features of video frames and a fast and flexible search algorithm named the max first search algorithm in order to realize a large-scale video frame retrieval system. This system navigates users to relevant frames in one or several videos according to their semantic preference expressed in the input image within the constrained response time.

In reality, most search algorithms run to completion: they provide users answers after performing some fixed amount of computation. However, different persons, or the same person under different circumstances, may have different interest under one query. In some cases, the users may wish to control actual response time while negotiating quality in return of search results. Traditional human factors research has also shown the need for response times faster than one second [1], or other usability studies suggest 8-

---

<sup>1</sup>Corresponding Author: Diep Thi-Ngoc NGUYEN, Faculty of Information Technology, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam; E-mail: ngocdiep@vnu.edu.vn.

second rule [2]. Nevertheless, there is no any fixed rule for response times. One user may be dissatisfied for search responses slower than one second but in some other cases, the same user may be willing to spend more time, even several hours, or days to get better answers to a very important search. There is also an equally important case in which the adaptive response time is on demand. Imagine an assembled system with a search engine, the system has a processing plan for its modules and needs some results of the search engine as input to the next module. For those reasons, the computation time of the search engine is preferred to be controllable.

In addition to time constraint preference, it is a semantic preference of users when searching for information. For example, one person may be more interested in the colors while another may be more interested in the shapes appears in the image query then evaluate the results of search as “relevant” or “irrelevant” to their intention. In more concrete situations, one person may be interested in some specific colors or some specific textural patterns in the image query. In more complex cases, the query can be created according to the imaginations of users using method introduced in [3]. To deal with this kind of dynamic changing contexts, each video frame and the input image are represented by set of low-level features, so-called metadata features, of color, shape, texture, etc. . . Tracking of as many semantic features as possible brings computer system closer to high-level concepts of data presented in human visual system. However, it leads to the creation of a very high dimensional search space in which dynamic query is complex and causes difficulties to pre-clustering or tree-based search algorithms. Obviously, if one input image is represented as a set of  $n$  features, the total number of possibly generated queries which reflect different contexts from the image is the number of distinct  $k$ -feature subsets given by the binomial sum:

$$\sum_{k=1}^n \binom{n}{k} = 2^n - 1 \quad (1)$$

The multimedia data we have today is exploding continuously and critically not only in volume (amount of data) but also in velocity (producing rate). Among multimedia data such as text, image, audio, and video, video data plays a major role in the rapid growth in size of big data. One text page paper might be 20KB whereas a high-definition video (1080i and 1080p uncompressed HDTV RGB(4:4:4)) is about 200MB per second. This means each second of video generates more than 10,000 times as many bytes as required to store one text page. Additionally, the video data is also increasing with high velocity due to current development of electric devices, mobile devices, and network infrastructure. Monitoring camera devices also become commonly used, set up in many locations, and activated as 24/7 monitoring services for environmental monitoring, or inspecting. Such camera devices generates a great amount of video data by time for each location while all of monitored video data is required to store time by time for further investigation. If video data is compressed into MP4 files, a compressed 1-minute MP4 file with medium quality needs 26MB to store. Video data from one camera for one day monitoring (24 hours) is 37.4 gigabyte, and for one month is over 1 terabytes. The captured data for one year from one device is about 13.5 terabytes. Therefore, the accumulated amount of data multiplied by the number of devices can petabytes, or more. Comparatively, user-generated video data is also getting bigger while easily recognizing that users apparently prefer to capture and share information via videos. According to

YouTube Statistics<sup>2</sup>, “100 hours of video are uploaded to YouTube every minute” which is approximately 156GB of data (compressed with medium quality).

Video data with its plentifully visual and motional content has been becoming a main source to explore knowledge. However, conventional video retrieval systems are mostly query-by-keywords rather than query-by-content, and mostly finding relevant videos rather than relevant frames in videos. Most of the video data are on the wild form without any text descriptions that handicap text-based retrieval. If only videos are retrieved, users still have to spend a lot of time to seek to relevant portions in the videos and this task might be very hard if the videos are long. Sivic et al. [4] introduced a method to find the objects which occur throughout one video. However, it will become less useful when users don’t know a specific video to be searched. Therefore, it is more preferable to users that they can directly go to scenes of interest in not only one video but several unknown videos and are able to watch the videos from those points.

The following sections in this paper discuss some prior researches relating to large-scale video indexing and retrieval, and describe our improved inverted index method to index video frames and the proposed max first search algorithm in details. We present some experimental studies based on a prototype video frame retrieval system including some intensive discussions in order to clarify the performance and potential of our proposed algorithm to scale to very large video data.

## 1. Indexing and Retrieval in Large-scale Datasets

Searching in multimedia data in which datasets are very large and their semantic features are represented in high dimensions has been main objective of many researches. There is a phenomenon so-called “curse of dimensionality” that arise when analyzing data in high-dimensional spaces. This phenomenon suggests that close objects might get separated by a partition boundary when partitioning the space. Many researches have utilized approximate nearest neighbors problems and distributed computing methods in order to solve large-scale multimedia retrieval task. Recently, content-based image retrieval systems can now manage collections having sizes that could be very difficult years back. Most systems can handle several million to hundred million images [5,6,7,8,9,10], billions of descriptors [11,8,12], or address web-scale problems [13,14,10,15], and so on.

### *Cluster-based search*

Many approximate high-dimensional near neighbor search methods are based on some kinds of segmentation of the data collection into groups named *clusters*, which are stored together on disk. At query time, an index is typically used to select the single nearest cluster for searching. The work [8] used cluster-based indexing method named the *extended Cluster Pruning* (eCP) proposed in [16] that is an extended method of *Cluster Pruning* (CP) [17]. The *Cluster Pruning* method [17], which is a simple algorithm comparing to traditional k-means algorithm, randomly chooses a subset of data points to be *leaders* and the remaining data points are partitioned by which leader is the closest. The eCP by Gudmundsson et al. gives some changes to improve CP by three additional parameters to control cluster size on disk, balance cluster size distribution in order to improve search

<sup>2</sup><http://www.youtube.com/yt/press/statistics.html>



in both IO and CPU costs. Other derivatives of cluster-based methods can also use *hierarchical clusters* and so-called *multiplelevel clustering* [16] or *multi-index* in [15] to partition large clusters into smaller clusters. The very recent work in very large scale image search [8] used clustering concept adapting with distributed Map-Reduce programming paradigm, and tested with Hadoop framework by indexing 30 billion SIFT descriptors for roughly 100 million images (about 4 terabytes of data). However, the search does unlikely support realtime application, and the quality of search was not mentioned in the paper.

The cluster-based indexing and retrieval methods can perform well for large-scale image retrieval, however they are less flexible in dynamic environment that supports dynamic selection of features according to users' preferences. In other words, handling dynamic queries at real-time search may require re-clustering for every entire dataset.

#### *Locality-sensitive hashing*

Some researches approach to high-dimensional image search for large datasets using "locality-sensitive hashing" (LSH), which is a solution for nearest neighbor problem such as [18,15]. The first locality-sensitive hashing algorithm was introduced very early in 1998 [19] to overcome "curse of dimensionality". A LSH algorithm uses a family of locality-sensitive hash functions to hash nearby objects in the high-dimensional space into the same bucket. A similarity search is performed by hashing a query object into a bucket, using the data objects in the bucket as the candidate set of the results, and then ranks the candidate objects using the distance measure of the similarity search. The goal of LSH is to maximize probability of "collision" of similar items rather than avoid them such like perfect hashing techniques. Some methods such as [18] improved space efficiency of LSH using *multi-probe* by deriving probing sequence to look up multiple buckets that have a high probability of containing the nearest neighbors of a query object.

Locality-sensitive hashing is similar to cluster-based algorithms since datasets are pre-indexed into groups and at search time, a group will be called providing candidates for similarity calculation. Similarly, there is still a need to improve hashing to adapt dynamic environment.

#### *Tree-based techniques*

Lejsek et al. in [7,11] introduced a *Nearest-Vector-tree* data structure named NV-tree for approximate search in very large high-dimensional collections. Another author, Liu in [20] used a distributed hybrid *Spill-tree*, a variant of the *Metric-tree*, for a collection of 1.5 billion global descriptors. When comparing to LSH, the performance was the same but the method used fewer disk reads. A NV-tree is constructed after a repeated steps of projection and partitioning through the high-dimensional space. The search algorithms using NV-tree mainly depend on the selected projection lines. Each projection line can be seen as a concrete context to be search. Due to a limited number of projection lines and their fixed contexts, the tree-based techniques are restricted to apply for dynamic queries.

The above discussed methods have some advantages for realizing high performance retrieval of large datasets, but they have not supported dynamic queries. In this paper, we design an indexing method that copes with high dimensional metadata of large datasets and at the same time deals with dynamic queries and accelerates searching.

## 2. Inverted Feature Indexes

An optimized data structure can facilitate efficient retrieval. In the document retrieval field, the speed of answering a query “find the documents where word X occurs” can be improved using *inverted indexes*. An inverted index (as known as *postings file* or *inverted file*) is structured like an ideal book index. It has an entry for each *word* (or *term*) in the corpus followed by a list of all the documents (with or without augmented data such as position in that document) in which the word occurs. When querying one or more terms, documents are retrieved by looking up indexes of corresponding terms, and they are processed by computing their vectors of word frequencies and ranked to return as closer distance to the query. The time and processing resources to perform the query is dramatically improved and apparently like with no-delay.

The idea of using an inverted index structure for image retrieval tasks is at least 10 years old, e.g., [21]. Researches in this direction treat either physical features of images such as [21] or “visual words” of quantized descriptors such as [4] as set of terms to be indexed. By searching time, the algorithms get out all the images which contain features appear in query and adding them to the pool of candidate images to be ranked. Some methods such as [4] used Matlab sparse matrix to keep inverted indexes in-memory, but this is not scale to size of datasets. As the story of big data opens, a disk-based inverted index should be considered. Hare et al [5] based on the work of Sivic et al. [4] proposed a platform for scalable high-performance image retrieval.

Considering inside structures of the inverted indexes, there are two common way to organize the indexes: *document-sorted* [4,5] and *frequency-sorted* as stated in [22]. Inverted indexes are generally document-sorted, that is, sorted by document identifiers. It is obvious to see that, using document-sorted indexes, whole of each list has to be pulled out for processing whereas using frequency-sorted indexes, the identifiers of the interesting documents might be brought to the start of the list. Therefore, frequency-sorting yields a reduction in disk traffic because only part of each inverted list must be retrieved. Persin [22] used frequency-sorted indexes although his work was targeted to document retrieval. A work of Miyagawa et al. [23] used frequency-sorting inverted indexes for image databases although they did not clearly state that. Comparing to the use of thresholds to decide one document is candidate or not [22], [23] used a more dynamic scheme to get candidates by checking similarity of current pulled document with previous document to decide which index should move to to get next candidates.

**Table 1.** Categories of methods use inverted index. The bold categories are used or newly proposed in this research

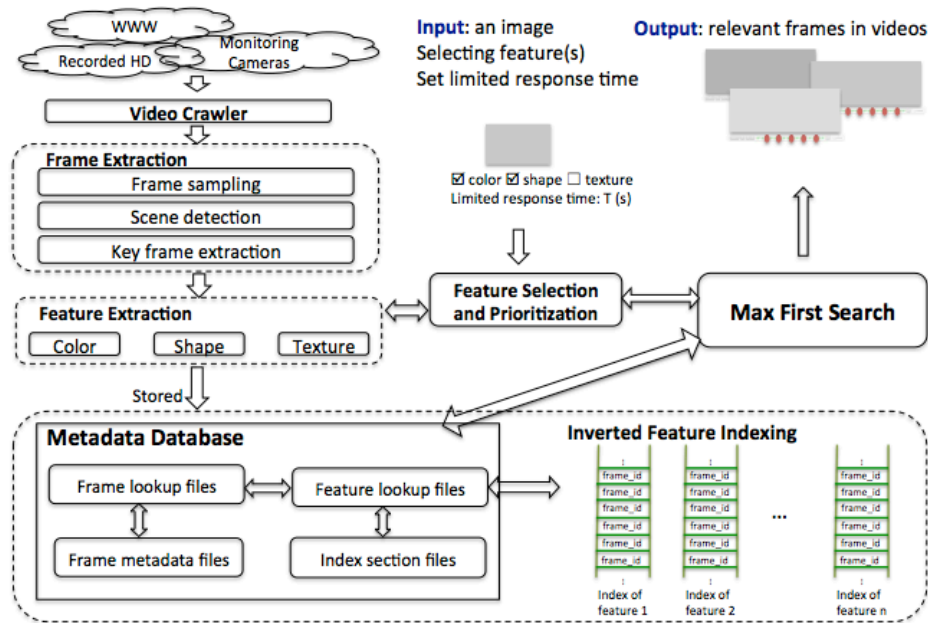
Viewpoint	Categories
Organization	in-memory, <b>disk-based</b>
List structure	document-sorted, <b>frequency-sorted</b>
Application	text retrieval, image retrieval, <b>video retrieval</b>
Content-based representation	<b>direct descriptors</b> , “visual words”
Search scheme	threshold-based, <b>dynamic path finding</b>

Table 1 summaries categories of methods which take advantage of inverted index from some viewpoints: organization of inverted index (either in-memory or disk-based),

structure of inverted index (either document-sorted or frequency-sorted), representation of semantic content (either using direct descriptors or quantized them into visual words), and its applications (either text, image or video retrieval). We present in this paper a disk-based, frequency-sorted inverted indexing method, which is as an improved method to support video retrieval by dynamic path finding search scheme. A newly designed search algorithm is named max first search algorithm.

### 3. Framework of Video Frame Retrieval System

Our video frame retrieval system using inverted feature indexes and max first search algorithm has overview architecture shown in figure 1. The system includes seven main modules: video crawler, frame extraction, feature extraction, feature indexing, database manager, feature selection and prioritization, and max first search.



**Figure 1.** Overview architecture of Video frame retrieval system with seven main modules: video crawler, frame extraction, feature extraction, feature indexing, database manager, feature selection and prioritization, and max first search.

The video crawler module either collects video urls from the Internet (e.g., from YouTube) or converts videos in local hard disks into MPEG-4 format. The frame extraction module contains two main functions, sampling video data to get frames, then applying scene detection method to extract representative frames for the video. The details of this module are discussed in section 4.1 and section 4.2. The feature extraction module uses some methods to extract descriptors of low-level features such as color, shape, texture, or compact features as combination of them. The feature indexing module creates indexes for videos, video frames and descriptor indexes. Those indexes are stored in files

which are managed by the database manager module. These two modules in implementation are integrated. This feature selection and prioritization module interprets the input image, features of interest and input limited time bound as users' context, then select corresponding feature indexes and prioritize them for search phrase. This max first search module implements the max first search algorithm in order to quickly find candidates and return answers when time bound is reached.

## 4. Video Frame Extraction

### 4.1. Video Sampling

A video is basically a sequence of time varying images. We can temporally sample the video to obtain  $R$  frames in one second. At the beginning of a video, we capture a first video frame, then for every  $\Delta_t = 1/R$  (second) we capture one frame from the video. The sampling time  $\Delta_t$  should representatively summary the content of the video in order to reduce redundancy when indexing. In other words, the captured video frames describe different scenes along the video sequence.

Nowadays, almost existing videos are compressed using MPEG-4 format or they are easily converted to MPEG-4 format using one of many available softwares available. In MPEG, each video sequence is divided into one or more groups of pictures (GOPs). Each GOP is composed of one or more pictures and starts with an I picture (*Intra coded picture*). The distance between two consecutive I pictures is referred to as  $N$ . Because an I picture provide a random access points to the compressed video data and it can be decoded independently without referencing to other pictures, the I picture is also called key frames of the video. For that reason, we suggest that video frames of a video should be extracted using GOPs with a sampling time as following:

$$\Delta_t = \frac{1000000}{\frac{R}{N}} \text{(microseconds)}, \quad (2)$$

where  $R$  is frame rate (frames per second),  $N$  is number of pictures in a GOP.  $R$  and  $N$  information are encoded in the header of the video.

For example, a video is created using  $R = 24$  frames for every second and compressed using MPEG with  $N = 12$ , which means two consecutive I pictures are separated in a 12-frame interval of frames. By sampling every  $\Delta_t = 1000/(24/12) = 500000$  microseconds starting from the beginning of the video, we can obtain key video frames of the video. In other words, in stead of extracting 24 frames every second, only 2 frames are extracted every second with our assumption that they are enough information about the video. However, extracting two I-frames per second still produces a large number of frames to be indexed and by observing that one frame per one second apparently appear as different scenes. Therefore, in practice, we sample videos by every second starting from the beginning of the videos.

### 4.2. Representative Frame Extraction using Two threshold-based Scene Detection

After sampling the videos, a scene detection algorithm is applied in order to keep only representative frames of each video. Scene detection which is also known as scene seg-

mentation is very significant for summarizing content of videos data and is considered in many research works such as [24,25,26]. Scenes can be determined based on motions in video and there are also many techniques to detect motions. However, when a fast analysis method is required to be suitable for real-time video indexing, threshold-based techniques may be a proper choice. In order of reducing penalty rate of missing a scene, we propose a scene detection algorithm using two semantic thresholds and one temporal threshold.

The scene detection algorithm detects new scene and keeps only the starting frame in the consecutive frames of a scene as a representative frame for the scene. The frame at 0 second (the beginning of the video) is the first presentative frame of a video by default. A frame is said to be on new scene if it is less similar to the current frame by a fixed threshold. We use cosine distance as similarity calculation which is defined using a dot product and magnitude as following for given two histograms,  $A$  and  $B$  with  $n$  elements:

$$similarity(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (3)$$

The computed distance ranges from 1 to 0 in which 1 means two frames is *completely similar* to other on computation, and 0 means two frames is *completely independent*.

Our proposed algorithm is based on an observation of the usefulness of the temporal distance between two frames. It is: two frames separated in a long time points in the video are likely in different scenes despite the similarity distance is not clear. This is because comparing similarity between two frames using their visual information presentation by low-level features is only relative.

---

**Algorithm 1** Advanced scene detection algorithm with two thresholds

---

```

1: Initialize: two thresholds  $DT$ ,  $CT$  and one passing time  $PT$ 
2: StartSceneFrame  $\leftarrow$  firstFrame
3: for all Frame in video sequence do
4:   if Similarity (StartSceneFrame, Frame)  $<$   $DT$  then
5:     new scene is detected
6:     StartSceneFrame  $\leftarrow$  Frame
7:   else
8:     if Temporal distance (Frame -StartSceneFrame)  $>$   $PT$  AND Similarity
       (StartSceneFrame, Frame)  $<$   $CT$  then
9:       new scene is detected
10:      StartSceneFrame  $\leftarrow$  Frame
11:     end if
12:   end if
13: end for

```

---

In our scene detection algorithm 1, we use two thresholds, one is called a *direct threshold* ( $DT$ ) and another one is called a *conditional threshold* ( $CT$ ) with  $CT > DT$ .  $DT$  is regardless to temporal distance (*passing time* ( $PT$ )) of two frames, while  $CT$  provides a delay step to check the temporal distance is whether greater than  $PT$ . These

two thresholds ( $CT$  and  $DT$ ) are chosen with following criteria: if semantic difference between frames in a video are not clear, the thresholds should be set with large values; if ones want to keep a lot of frames for indexing, smaller values should be set. The  $PT$  threshold should be set large if ones don't want to pass many frames. Besides,  $CT$  is constrained to be larger than  $DT$ . Experimental studies shows good thresholds for common video datasets are  $DT = 0.76$  and  $CT = 0.91$ . The passing time  $PT = 5$  (seconds).

We realized that selecting features for calculating similarity depends on types of the video dataset. For example, if the video dataset consists of movies, the change between scenes in color is smaller than in shape, thus, using only color feature is not effective to detect scenes. In other hands, if the video dataset consists of animation or includes slideshows of flowers, color feature is apparently more effective than other features. In this paper, we use the CEDD descriptor (described in section 4.3 for both color and edge directivity features in intention to deal with a mixed video dataset. Besides, CEDD descriptors can be extracted fast and with small size. These advantages are promising for real-time video indexing.

### 4.3. Feature Extraction

We extract three kinds of low-level features from each frame that are color, shape, and texture. This paper does not focus on proposing a new feature extraction method so that commonly used extraction methods in literature are chosen. However, there are three criteria when choosing such methods: (1) extracted features are small in size, (2) requiring low computational power (in order to make the system suitable for real time indexing), and (3) descriptors of features generated by the methods must be on statistical representation scheme not structural representation scheme. We chooses six kinds of features which are: HSB colors, PHOG, CEDD, FCTH, and JCD to extract content of video frames.

Color features are extracted to 63-bin HSV color histogram by splitting HSB color space in a non-uniform way ( $7 \times 3 \times 3$ ) [27]. Shape features are extracted based on the spatial distribution of edges as "Pyramid of Histograms of Orientation Gradients (PHOG)" vector representation [28]. The shape feature PHOG vector has 40 elements while using two levels and 8 orientations. The integrated feature vector of color and texture is FCTH, which is a fuzzy color and texture histogram introduced by Savvas et al. [29]. This method uses a two-input fuzzy system to generate 24-bin color histogram, then uses Haar wavelet transformation for fixed 8 regions to export texture elements, and consequently total  $8 \times 24 = 192$ -bin FCTH feature vector is extracted as a packed feature of both color and texture features. This feature is chosen because of its robustness to deformations and noise. A compact feature vector of both color and edge directivity are introduced in [30]. Similar to FCTH feature, this method uses a two-input fuzzy system to generate 24-bin color histogram, then applies to a set of 6 texture filters which contains 5 digital filters of MPEG-7 edge histogram descriptor: vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edges. It is to note that the sixth filter is for filtering *no edge region*. Consequently, the CEDD histogram includes  $6 \times 24 = 144$  elements. The JCD feature vector is a joint descriptor joining CEDD and FCTH descriptors [31]. The joint JCD histogram includes 168 elements in total.

## 5. Video Indexing using Improved Inverted Feature Indexes

### 5.1. Design Principles of Indexing and Storing

The indexing method for video frames is based on two principles: (1) each feature is independently indexed, and (2) each index keeps a list of frame identifiers in descending order of the corresponding values of the feature of the frames. Such indexing is general in inverted indexing techniques as described in section 2, in which the ‘*term*’ is equivalent to a metadata ‘*feature*’ extracted from video frames and the ‘*document*’ is equivalent to a video frame. Concretely, if a frame has a unique identifier denoted as  $frame_i$ , its metadata is expressed as a row vector of  $n$  features  $f_1, f_2, \dots, f_n$  with corresponding values  $K_{f_1}^{(i)}, K_{f_2}^{(i)}, \dots, K_{f_n}^{(i)}$ . The inverted index of each feature  $f_j$  is denoted as  $fi_j$  and keeps lists of frames. The frames in an index are always sorted by decreasing ordered of their corresponding values of the feature. If a permutation  $\varphi_j(1)\varphi_j(2) \dots \varphi_j(N)$  of  $N$  video frames represent that order in the  $j$ -th index, the expression of feature indexes is as following:

$$\begin{array}{ll}
 \text{\textit{N frame indexes}} & \text{\textit{n feature indexes}} \\
 \\
 frame_1 = \{K_{f_1}^{(1)}, K_{f_2}^{(1)}, \dots, K_{f_n}^{(1)}\} & fi_1 = \{frame_{\varphi_1(1)}, frame_{\varphi_1(2)}, \dots, frame_{\varphi_1(N)}\} \\
 frame_2 = \{K_{f_1}^{(2)}, K_{f_2}^{(2)}, \dots, K_{f_n}^{(2)}\} & fi_2 = \{frame_{\varphi_2(1)}, frame_{\varphi_2(2)}, \dots, frame_{\varphi_2(N)}\} \\
 \vdots & \vdots \\
 frame_N = \{K_{f_1}^{(N)}, K_{f_2}^{(N)}, \dots, K_{f_n}^{(N)}\} & fi_n = \{frame_{\varphi_n(1)}, frame_{\varphi_n(2)}, \dots, frame_{\varphi_n(N)}\} \\
 & \text{where } K_{f_j}^{(\varphi_j(1))} \geq K_{f_j}^{(\varphi_j(2))} \geq K_{f_j}^{(\varphi_j(N))}
 \end{array}
 \implies$$

It is to note that each feature has its own index regardless of its kind, in other words, even the features are in one kind (e.g., HSB color, PHOG, CEDD, etc.) they have their independent indexes. When a new video frame comes to the database, it is inserted into every  $n$  feature indexes at proper positions such that each index is always ordered as well. Although this kind of insertion strategy make inserting process slow, the data structure dramatically empowers searching process as shown in the experimental studies section.

All feature indexes are stored in disk. In addition, data are written by bytes to files in order to enable random access which fastens reading data at arbitrary locations out from the database without reading from the beginning of the files.

### 5.2. Database Structure

Based on above design consideration, a database structure is constructed to store metadata information of video frames and inverted indexes of metadata features in files. Currently, we don’t focus on database buffering, query plans, and logging but storage strategy. There are four types of files used in the database: frame lookup (*.pi* files), frame metadata (*.bu* files), feature lookup (*.t* files), and index section (*§* files).

#### *.pi* files

The frame lookup files (*.pi* files) are the look-up tables of frames that contain ranges of frame identifiers (ID) and corresponding metadata files in which the metadata information of the frames are stored.

- Filename format: [db\_name].pi  
For example, *50TB\_Animation.pi* means database name is “50TB animation”.
- Format of each line: frame\_start\_id frame\_end\_id .bu\_file  
For example, *1 100 256272829292* means that metadata of frames having ID from 1 to 100 is stored in *256272829292.bu* file. Normally, the end ID should be bigger than start one.

#### *.bu files*

The frame metadata files (*.bu* files) are data files containing in each line the metadata of one frame, such as video ID in which the frame is in, file name of the thumbnail of the frame, and list of features, etc.

- Filename format: [serial].bu  
This filename is a random long number, for example, *256272829292.bu*
- Format of each line: frame\_id frame\_name list\_of\_metadata  
For example, *19 TeddyBear\_210 2.1 1.0 0 2.9 0 0...* means that the 19th frame in the database belongs to TeddyBear video.

#### *.t files*

The feature lookup file (*.t* files) or tree files are look-up tables of feature indexes. They contain the value ranges of the features and corresponding index section files.

- Filename format: [descriptor\_id].t  
For example, *Color\_19.t* means that this file contains indexing of the 19th descriptor of Color feature.
- Format of each line: value1 value2 .s\_file  
For example, *4.123 6.345 1273733990202* means values of all frames in *1273733990202.s* file range from 4.123 to 6.345

#### *.s files*

The index section files (*.s* files) keep identifiers of frames in corresponding interval values of the given feature.

- Filename format: [serial].s  
This filename is a random long number, for example, *1273733990202.s*
- Format of each line: value frame\_id  
For example, *4.512 123* means the value of frame ID 123 is 4.512

### 5.3. Frame Insertion

When a frame is inserted to the database, at first its features are extracted. Then, it will request for an ID in the database. This ID is unique by setting it equal the maximum ID plus one. Location of *.bu* file is detected in *.pi* file based on this ID. If this ID is out of ranges in the *.pi* file, a new range containing this ID is added and a new *.bu* file is generated. The new range is defined as a constant, such as length of 1000 or 10000 data. Metadata of this ID is then written in that *.bu* file.



In the next step, the features are indexed and written to *.t* and *.s* files. Each feature will be sent to the corresponding feature lookup file. The location of *.s* file is decided based on the value of the feature. The position of this feature in the sorted list of the *.s* file is found by using binary searching.

#### 5.4. File Partitioning

The size of *.bu* and *.s* files becomes bigger when there are more frames inserted to the database thus a method for partition the files to smaller files is required in order to fasten later reading processes. A maximum number of frames which is also the number of lines in a *.bu* file is set, for example, 1000 or 10000 data. While indexing if this number is reached, a new *.bu* file is added.

Partitioning of *.s* file is more complex due to the sorted list of value. Similar to a *.bu* file, a constant of maximum length is used to divide into smaller files. The middle value of the sort list is chosen for division. It means that the first half will become a small file, and the second half is stored as another small file. Since one more *.s* file is added, its *.t* file is modified to store new ranges.

## 6. Max First Search Algorithm

### 6.1. Context-dependent Feature Selection and Prioritization

Given an input query is an image which has a metadata vector represented by  $Q^{(n)} = (K^{(1)}, K^{(2)}, \dots, K^{(n)})$  in which  $K^{(i)}$  is the value of the corresponding *i*-th feature. It is to note that this query might be created based on users' imaginations using some combining operations applied to two or more input images that have already proposed in [3]. According to the selected boxes for features of interest as shown in the input part in the figure 1, a set of proper features are chosen. We define a rule to choose features to be used: if 'color' is selected, then choose HSB color feature set; if 'shape' is selected, then choose PHOG feature set; if 'color' and 'shape' are selected, then choose CEDD feature set; if 'color' and 'texture' are selected; then choose FCTH feature set; and if all 'color', 'shape' and 'texture' are selected, then choose JCD feature set. As a result, we have a *k* features which forms a sub-space to search.

Assuming that the greatness in values of features affects the computed similarity over other features, the index of the feature whose value is greater than those of others will be traversed earlier than the others. In other words, after selecting *k* features according to users' intentions, we sort them by their values in descending order. Sorting *k* features is equal to deciding a permutation of *k* features which is  $q(1)q(2) \dots q(k)$  where:

$$K^{(q(1))} > K^{(q(2))} > \dots > K^{(q(k))} \quad (4)$$

After all, the generated query consists of *k* features is written as:

$$Q^{(k)} = (K^{(q(1))}, K^{(q(2))}, \dots, K^{(q(k))}) \quad (5)$$

The permutation  $q(1)q(2) \dots q(k)$  also determines priorities of descriptors while a search algorithm runs. Concretely, the descriptor  $q(1)$  with corresponding value  $K^{(q(1))}$  has the first priority, rather the descriptor than  $q(2)$  with corresponding value  $K^{(q(2))}$ , etc., and the descriptor  $q(k)$  with corresponding value  $K^{(q(k))}$  has least priority.

## 6.2. Max First Search Algorithm

The max first search algorithm repeatedly finds a candidate frame  $A$  which is projected to subspace formed by  $k$  selected features as  $A^{(k)} = (K_A^{(q(1))}, K_A^{(q(2))}, \dots, K_A^{(q(k))})$  while maximizing  $similarity(Q^{(k)}, X^{(k)})$  which is either cosine distance as defined in formula (3) or inner product distance as defined in formula (6) depends on searching intentions. The inner product distance of two equal-length vector  $X$  and  $Y$  is the sum of the corresponding bins as following:

$$similarity(X, Y) = X \cdot Y = \sum_{i=1}^n X_i \times Y_i \quad (6)$$

The inner product is effective when searching among specific indexes with intentions of exploring those dominant features. This is based on a property of inner product: the sum  $\sum_{i=1}^n X_i \times Y_i$  will be large if each value  $X_i$  and  $Y_i$  are large. When the values of the query, e.g.  $X_i$  does not change, the sum will be large if the corresponding values of the candidate frame are large. This property involves our expectation to find answers which have high values of features of interest. For example, even if the input query has small distribution of red colors, users prefer the red colors and suppose to get answers which have large red colors. On the other hand, the cosine distance is effective when searching for similar frames.

The outline of max first algorithm is sketched in algorithm 2. The algorithm repeatedly checks vertical and horizontal candidates starting from the top frame of the first prioritized index. It changes to the next prioritized feature index when a better similarity is found at that index. The algorithm finishes running whether the input limited time bound is elapsed.

It is important to realize that in practice, a *local penalty* can occur. A local penalty is defined by a following example. In the current feature index  $\tau$  with current checking frame  $A_i$ , if frames from the top of the index are listed as

$$A_0, A_1, \dots, A_{i-1}, A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots$$

with, of course, their corresponding values are in descending order

$$K^{(A_0^{(\tau)})} \geq K^{(A_1^{(\tau)})} \dots \geq K^{(A_{i-1}^{(\tau)})} \geq K^{(A_i^{(\tau)})} \geq K^{(A_{i+1}^{(\tau)})} \geq K^{(A_{i+2}^{(\tau)})} \geq \dots$$

It is said that a local penalty occurs in the index  $\tau$  at the frame  $A_i$ , for instance, when  $similarity(A_i^{(k)}, Q^{(k)}) < \xi < similarity(A_{i+1}^{(k)}, Q^{(k)})$ . In other words, a local penalty occurs when better answers can not be reached because a break is made at local candidate in the index. Such local penalty can be resolved using a delay parameter, that waits for a number of candidates to be checked before not finding any better answers.

---

**Algorithm 2** Max first search algorithm. Iteratively find a candidate for similarity calculation based on dominant features

---

```

1: [Initialize check list] Initialize a NeighborSimilarity list
2: [Initialize result list] Initialize a Results list
3: Move to first prioritized descriptor index
4: [Get first candidate] Fetch first frame at the top of the current index
5: [Similarity calculation] Compute similarity with query
6: [Push to result list] Results.push(current fetched frame)
7: repeat
8:   [Get vertical neighbor candidate] Fetch unchecked frame from the top of the
   current index
9:   [Similarity calculation] Compute similarity with query as  $\sigma$ 
10:  [Push to result list] Results.push(current fetched frame)
11:  [Get horizontal neighbor candidate] Fetch unchecked frame from the top of the
   next prioritized index
12:  [Similarity calculation] Compute similarity with query as  $\xi$ 
13:  [Push to check list] NeighborSimilarity.push( $\xi$ )
14:  [Push to result list] Results.push(current fetched frame)
15:  if  $\sigma <$  any of  $\xi \in$  NeighborSimilarity then
16:    if current index is the last index then
17:      [Round change index pointer] Move to the first prioritized index
18:    else
19:      [Change index pointer] Move to the next prioritized index
20:    end if
21:    [Reset check list] NeighborSimilarity.clear()
22:    [Loop] Back to 8
23:  else
24:    [Loop to lower] Back to 8
25:  end if
26: until time bound is elapsed or interrupt event is raised
27: [Ranking] Sorting Results list by descending order of similarity
28: [Display] Output to outputstream

```

---

## 7. Experimental Studies

### 7.1. Video Dataset

For experimental studies, a dataset of 2.04TB video is prepared. The dataset is stored in an external hard disk and contains 407 videos of action, fantasy, comedy, adventure, documentary kinds. The total length of videos is 453.24 hours. The system are implemented running on a desktop computer of 4 Core i7, 3.1 GHz CPU and 16GB Memory, configured with Ubuntu 13.04 operating system. The six kinds of features for color, shape and texture features which are HSB color, PHOG, Gabor, CEDD, FCTH, JCD using an open source Java library JFeatureLib [27]. The total number of extracted frames is 277,288 and the total size of their metadata is 1.6GB. The time to extract frames and their metadata of one-hour video is averagely 2 minutes, and the average time to extract full metadata features of one frame is 1.0 second.

In order to examine the algorithm from various aspects, the dataset is divided into smaller datasets by the number of frames which are 10000, 50000, 100000, and 200000 datasets. The indexing time, the size of generated database, the number of indexing files of each dataset are shown in table 2.

**Table 2.** Indexing time, size of generated database, number of indexing files of each video dataset

Dataset (number of frames)	10000	50000	100000	200000
Indexing time (minutes)	7	84	274 ( $\approx$ 4.5 hours)	929 ( $\approx$ 15.5 hours)
Size of indexes	224MB	1.2GB	2.3GB	4.6GB
Number of <i>.pi</i> file	1	1	1	1
Number of <i>.bu</i> files	201	1000	2000	4000
Number of <i>.t</i> files	607	607	607	607
Number of <i>.s</i> files	49783	254767	516169	1048600

## 7.2. Response Time Limitation versus Quality of Search

A brute force searching is also implemented in order to get an “expected” ranking results. The brute force search wholly runs over the database, returns top 20 results with highest similarity to the input image. This experiment examines the effectiveness of the search algorithm by its results at different limited response time comparing with the expected results for some same input. Running a brute force search over a database returns an expected ranking list as  $S = \{S_1, S_2, \dots, S_{20}\}$  where  $S_j$  indicates the frame identifier at  $j$ -th rank in  $S$ . Running the max first search algorithm with a limited time  $T$  returns a ranking list  $R = \{R_1, R_2, \dots, R_b\}$  where  $R_i$  is the frame at  $i$ -th rank in  $R$  and  $b \leq 20$  is the number of returned frames if it is not greater than 20, otherwise  $b = 20$ . Our system may return a small number of results if it is constrained to run in small time  $T$ . In this case, the algorithm can only check some candidates then stops to return answers.

We introduce two measures to evaluate the effectiveness of the algorithm: success rate and deviation of ranks. The success rate  $\varsigma$  measurement defines the percentage of expected results that can be retrieved using proposed algorithm. It is:

$$\varsigma = \frac{|R \cap S|}{b} \times 100 (\%) \quad (7)$$

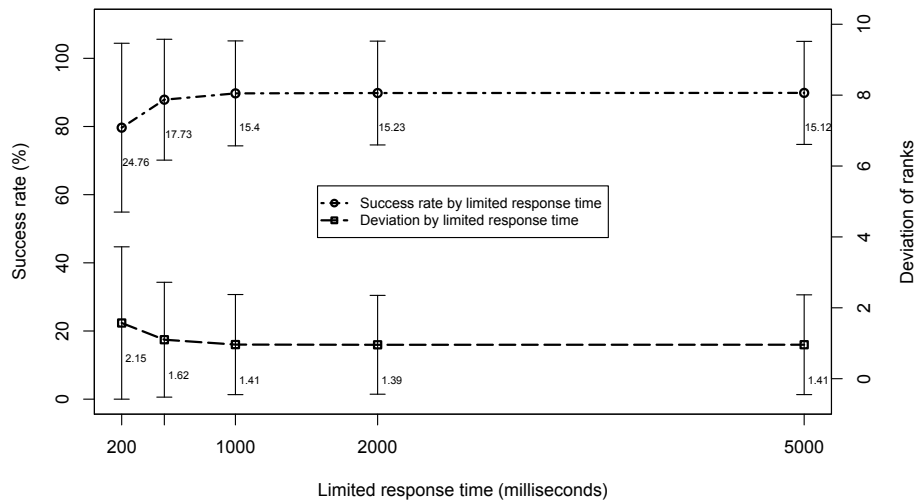
where  $|X|$  denotes the cardinality of the set  $X$ .

The deviation of ranks  $\delta$  measurement defines the difference between ranks of the frames in  $R$  list comparing to their ranks in the expected list  $S$  as in equation (8).

$$\delta = \sqrt{\frac{1}{|R \cap S|} \sum_{1 \leq i \leq b} (i - j)^2 \text{ subject to } R_i = S_j} \quad (8)$$

We randomly create 250 queries an run each query at five limited times of 0.2, 0.5, 1, 2, and 5s using the max first search algorithm and also run it using the brute force search. The generation of each query is: at first, randomly selecting an image, then, randomly

generating a number  $k$  which decides the number of features will be used, and at last, selecting a set of  $k$  features from each kind of feature (color, shape, texture). The inner product distance defined in formula (6) is used to compute similarity of a query and a target frame as suggested in section 6.2. To each query at a limited time, we compute  $\sigma$  and  $\delta$  and the results for 250 queries are summarized in figure 2.



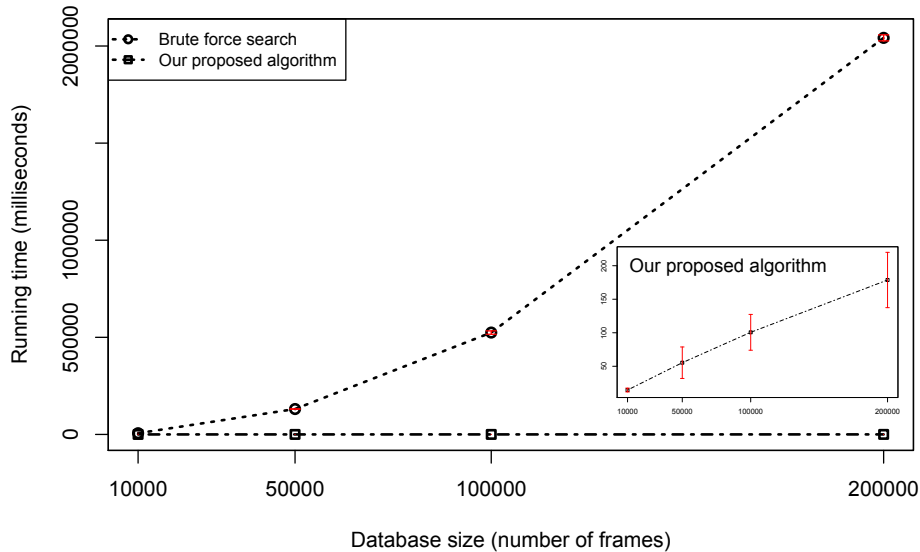
**Figure 2.** Success rate and deviation of ranks of results when comparing to brute force searching at different limited response time of 0.2, 0.5, 1, 2, and 5 seconds. (Using 50000-frame database)

The graph in figure 2 shows our algorithm averagely achieved high success rate (from 80%) even in small time limitation. The low line in the graph shows the deviation of ranks at different limited response time. A deviation of 1 means the results retrieved by our algorithm can be different one position with their expected ranks. The deviation of ranks is averagely decreasing that suggests our algorithm can retrieve better ranks by expanding time limitation. Although the standard deviation (SD) of the success rate and deviation of ranks at 0.2s are large, these SDs decrease when larger time limitation is given. This decreasing SDs mean the performance of our algorithm becomes stable when given larger time. The graph also shows that our algorithm is able to retrieve over 70% of expected results with slightly different ranks from 0.5s.

### 7.3. Running Time versus Size of Database

This experiment is for examining the scalability of our search algorithm comparing to the brute force search regarding the increasing size of database. We assume that when a half of good results is retrieved, the algorithm is considered as acceptable. In this experiment, we mark the running time of our algorithm when it already retrieves 50% of expected results, which are returned by the brute force search for a same given input. And this running time is the main standard to evaluate. We also randomly generate queries by the

generation method in section 7.2 and run the queries with 10000, 50000, 100000 and 200000 databases described in section 7.1. The running time regarding size of databases are plotted in figure 3.



**Figure 3.** Running time regarding size of database.

The graph in figure 3 shows that the running time of the brute force search increases quickly by size of the database. The average running time of the brute force search over 10000, 50000, 100000, and 200000-frame databases are 6, 130, 524, and 2042 seconds, respectively. In contrast, our algorithm could retrieve the half of the expected results in the average running time which is linearly increasing but still less than a second. Its running time for corresponding size of databases is: 0.014, 0.055, 0.1 and 0.18 second. This indicates the potential of our algorithm to be applied to large-scale database.

## 8. Conclusions

This paper has presented an adaptive search path traverse algorithm using an improved inverted indexes for metadata features of video frames. The algorithm is designed to deal with variously changing preferences of users in every query that is created by their imagination and limited by a preferred response time. The proposed algorithm priorities feature indexes according to input query and quickly finds an initial highest match-possibility candidate. Then it iteratively traverses over feature indexes to find other neighbor candidates until the input time bound is elapsed.

An additional advantage of using our improved inverted indexing is easy insertion to database. Adding new kind of features of video frames does not affect already indexed features. This advantage for vertical insertion by features is also promising for distributed indexing method.

The experimental studies based on video frame retrieval system show the feasibility and effectiveness of our proposed search algorithm that can return results in a fraction of a second with a high success rate and small deviation to the expected ones. In addition, the potential to be scalable to large dataset while preserving its search performance and the short indexing time make the video frame retrieval system able to work as a real-time video frame retrieval application for large-scale video dataset. Furthermore, the system is expected to bring users new video searching and watching experiences by its ability that it can navigate users to relevant frames in one or several videos according to their semantic preference expressed in the input image within the constrained response time.

## References

- [1] Jakob Nielsen. The need for speed. *Alertbox* (web page: <http://www.useit.com/alertbox/9703a.html>), 1997.
- [2] Andrew B King. *Speed up your site: web site optimization*. New Riders, 2003.
- [3] Diep Thi Ngoc Nguyen and Yasushi Kiyoki. An imagination-based query creation method for image retrieval. In *Information Modelling and Knowledge Bases XXIV*, pages 201–220, 2013.
- [4] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.
- [5] Jonathon S Hare, Sina Samangooei, David P Dupplaw, and Paul H Lewis. Imagerterrier: an extensible platform for scalable high-performance image retrieval. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, page 40. ACM, 2012.
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.
- [7] Herwig Lejsek, Friorik Ásmundsson, B Th Jónsson, and Laurent Amsaleg. Nv-tree: An efficient disk-based index for approximate search in very large high-dimensional collections. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):869–883, 2009.
- [8] Diana Moise, Denis Shestakov, Gylfi Gudmundsson, and Laurent Amsaleg. Indexing and searching 100m images with map-reduce. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 17–24. ACM, 2013.
- [9] Hervé Jégou, Florent Perronnin, Matthijs Douze, Cordelia Schmid, et al. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1704–1716, 2012.
- [10] Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: building a web-scale landmark recognition engine. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1085–1092. IEEE, 2009.
- [11] Herwig Lejsek, Björn Jónsson, and Laurent Amsaleg. Nv-tree: nearest neighbors at the billion scale. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 54. ACM, 2011.
- [12] Hervé Jégou, Romain Tavenard, Matthijs Douze, and Laurent Amsaleg. Searching in one billion vectors: re-rank with source coding. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 861–864. IEEE, 2011.
- [13] Matthijs Douze, Hervé Jégou, Harsimrat Sandhwalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, page 19. ACM, 2009.
- [14] Michal Batko, Fabrizio Falchi, Claudio Lucchese, David Novak, Raffaele Perego, Fausto Rabitti, Jan Sedmidubsky, and Pavel Zezula. Building a web-scale image similarity search system. *Multimedia Tools and Applications*, 47(3):599–629, 2010.
- [15] Xirong Li, Le Chen, Lei Zhang, Fuzong Lin, and Wei-Ying Ma. Image annotation by large-scale content-based image retrieval. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 607–610. ACM, 2006.

- [16] Gylfi Gudmundsson, Björn Jónsson, and Laurent Amsaleg. A large-scale performance study of cluster-based high-dimensional indexing. In *Proceedings of the international workshop on Very-large-scale multimedia corpus, mining and retrieval*, pages 31–36. ACM, 2010.
- [17] Flavio Chierichetti, Alessandro Panconesi, Prabhakar Raghavan, Mauro Sozio, Alessandro Tiberi, and Eli Upfal. Finding near neighbors through cluster pruning. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 103–112. ACM, 2007.
- [18] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *Proceedings of the 33rd international conference on Very large data bases*, pages 950–961. VLDB Endowment, 2007.
- [19] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.
- [20] Ting Liu, Charles Rosenberg, and Henry A Rowley. Clustering billions of images with large scale nearest neighbor search. In *Applications of Computer Vision, 2007. WACV'07. IEEE Workshop on*, pages 28–28. IEEE, 2007.
- [21] David McG Squire, Wolfgang Müller, Henning Müller, and Jilali Raki. Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. 1998.
- [22] Michael Persin, Justin Zobel, and Ron Sacks-Davis. Filtered document retrieval with frequency-sorted indexes. *JASIS*, 47(10):749–764, 1996.
- [23] Akiko Miyagawa, Yasushi Kiyoki, Takayuki Miyahara, and Takashi Kitagawa. A fast semantic associative search algorithm for image databases. *Transactions*, 41:1–10, 2000.
- [24] Behzad Shahraray. Scene change detection and content-based sampling of video sequences. In *Proc. SPIE*, volume 2419, 1995.
- [25] John R Kender and Boon-Lock Yeo. Video scene segmentation via continuous video coherence. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 367–373. IEEE, 1998.
- [26] Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. Video summarization and scene detection by graph modeling. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(2):296–305, 2005.
- [27] Franz Graf. Jfeaturelib – a free java library containing feature descriptors and detectors, <https://jfeaturelib.googlecode.com>, 2012.
- [28] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM, 2007.
- [29] Savvas A Chatzichristofis and Yiannis S Boutalis. Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on*, pages 191–196. IEEE, 2008.
- [30] Savvas A Chatzichristofis and Yiannis S Boutalis. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Computer Vision Systems*, pages 312–322. Springer, 2008.
- [31] S Chatzichristofis, Y Boutalis, and Mathias Lux. Selection of the proper compact composite descriptor for improving content based image retrieval. In *Proc. of the 6th IASTED International Conference*, volume 134643, page 064, 2009.
- [32] Henning Müller, Wolfgang Müller, David McG Squire, Stéphane Marchand-Maillet, and Thierry Pun. Performance evaluation in content-based image retrieval: overview and proposals. *Pattern Recognition Letters*, 22(5):593–601, 2001.



# Towards Finding Good Twitter Users to Follow Based on User Classification

Tomoya NORO, Atsushi MIZUOKA, and Takehiro TOKUDA

*Department of Computer Science, Tokyo Institute of Technology, Japan*

**Abstract.** In Twitter, finding good users to follow on a topic of interest is one way to collect, provide, and share information on the topic efficiently. However, it is not easy to find such users due to a massive number of users. In this paper, we classify Twitter users according to tweet frequency, tweet content, communication style, and follow relation, then present a method for finding good users to follow based on the user classification. The method is incorporated with our previous work on Twitter user search, and we show that the search accuracy is improved by the method.

**Keywords.** Social Network Analysis, Microblog, Twitter, Classification, Search

## Introduction

Twitter is getting more and more important platform of collecting, providing, and sharing information on a topic of interest. Although following users who provide valuable information on the topic is one common way to achieve an efficient information collection, it is difficult for us to find such users due to a massive number of users. We previously presented some methods for finding good users to follow for getting information about a topic of interest, such as a method considering tweet relation among users [1] and a method watching consistency in tweet content of each user [2]. In these methods, we assumed that good users to follow have the following characteristics: (1) they post many tweets and retweets on the topic of interest which are retweeted and replied to by many other good users to follow, and (2) they continuously post tweets related to the topic.

However, some other kinds of user characteristics could also be considered. For example, users who post tweets frequently and regularly may be better than the others, but users who post too many tweets all day are noisy and they may be inappropriate users to follow. Users who always provide new information would be better than users who repeatedly post similar tweets. Users who often interact with others may be good users to follow since we can expect that they will give us valuable information through interaction. Users who have only a few followers may not be good users since they should have more followers if they provide a lot of valuable information. We think the search accuracy could be improved if these aspects are considered.

In this paper, we classify Twitter users according to such tweet activity and user relation, then present a method for finding good users to follow in a list of users based on the user classification. In evaluation, we incorporate the method with our previous work and show that the presented method improves the search accuracy.

This paper is organized as follows. In section 1, we introduce some related works. We describe characteristics of good users to follow in section 2. We define Twitter user classes in section 3, then determine what features of each user are used for the user classification in section 4. We present a method for finding good users to follow in section 5. Evaluation results are shown in section 6, and we conclude this paper in section 7.

## 1. Related Work

Klout (<http://klout.com/>) used to provide the Klout Style. It classified SNS users into 16 categories by considering 4 aspects. This classification indicated each user's behaviour in social networks, but it was not designed for judging good users to follow.

Chu et al. classified Twitter users into 3 categories: human, bot, and cyborg [3]. They aimed at finding spammers. However, bots are not always spammers, and humans are not always good users to follow either. Other aspects also need to be considered.

Cha et al. investigated characteristics of Twitter users [4]. They said users who have many followers are popular but not necessarily influential, while users who are retweeted and mentioned many times have ability to post valuable tweets and ability to engage others in conversation respectively. It could be one aspect of user classification for finding good users to follow, but we think it is not enough.

Outside Twitter user classification, Broder et al. classified Web pages based on link structure, and showed bow-tie structure of the Web [5]. This idea could be applied to user relation graph on Twitter. However, we need to consider not only user relation but also other information such as tweet frequency, content, and so on.

## 2. Characteristics of Good Twitter Users to Follow

We think good Twitter users to follow have some of the following characteristics.

**Active, but not noisy:** Users who post tweets frequently will be good users to follow. However, users who post too many tweets will be noisy and they are not good users to follow.

**Tweeting regularly:** Users who post tweets regularly will be good users to follow although some of them may be spammers who post too many tweets.

**Non-repetitive:** Users who post similar tweets repeatedly may be spammers. Even if they are not spammers, they are not good users to follow since they do not provide new information.

**Informative:** Users who post many tweets valuable for others will be good users to follow.

**Attracting many other users:** Good users to follow are watched by many users since they provide a lot of valuable information.

**Frequently mentioned or interacting:** Well-known users frequently mentioned by others will be good users to follow. Users who often interact with others will also be good users since they will provide valuable information through interaction.

**Non-isolated:** Users who have only a few followers will be inappropriate users. They should have more followers if they are good users to follow.

In order to judge if each user has some of the characteristics, we define some Twitter user classes according to tweet activity and user relation in the next section.

### 3. Twitter User Classes

#### 3.1. Twitter User Classes According to Tweet Activity

We define 8 user classes based on 4 aspects in tweet activity such as tweet frequency, interval and content.

**Active/silent users:** The active users post tweets frequently while the silent users do not. The silent users may only be watching other users' tweets or they have already retired.

**Regular/occasional users:** The regular users post tweets regularly while tweet interval of the occasional users is not stable.

**Repetitive/random users:** The repetitive users post similar tweets repeatedly, while the random users usually post something new.

**Informative/chatting users:** The informative users post many tweets valuable for others, while the chatting users mainly post private tweets for greeting to friends (e.g. "Hello"), monology (e.g. "It's cold"), and so on.

#### 3.2. Twitter User Classes According to User Relation

We define 9 user classes based on 3 aspects in user relation such as interaction by retweeting/mentioning and following.

**Attracting/ignored users:** The attracting users are watched by many users, while the ignored users are not.

**Outgoing/authorized/interactive/lonely Users:** The outgoing users often mention other users, while the authorized users are often mentioned by others. The interactive users engage in conversation with others. The lonely users have little interaction with others. Well-known people may be the authorized users while some of them are not interactive.

**Popular/mutually-connected/isolated users:** The popular users are followed by many users, and the mutually-connected users have many mutually-following friends. The isolated users have only a few followers.

### 4. Features for Twitter User Classification

We consider 12 features to classify Twitter users.

**Average tweet interval, and coefficient of variation (CV) of tweet interval:** Average tweet interval of the active users will be small, and its standard deviation will be small in the case of the regular users. If average tweet interval is large, its standard deviation also tends to be large. In order to normalize the situation, we also consider coefficient of variation (CV) of tweet interval.

**Duplicate tweet rate:** In the case of the repetitive users, duplicate tweet rate among their tweets will be large. Some users repeatedly post the same text with different user mentioning, and some other users post the same URL with different text. We consider not only duplicate of entire tweets but also duplicate of tweets after removal of entities (URLs, hashtags, and user mentions) and duplicate of expanded URLs.

**Average tweet length:** Longer tweets have more information. Neubig et al. said, while the language of retweeted tweets tends to be more consistent, retweeted tweets tend to be longer and thus contain more information on the whole [6]. According to this observation, we take average length of entire tweets and average length of tweets after removal of entities.

**Times retweeted, retweeted tweet count, and H-index of times retweeted:** Tweets of attracting users will be retweeted many times. We also consider the number of retweeted tweets (tweets retweeted at least once) and H-index of times retweeted [7] to distinguish between users who post many tweets retweeted only a few times and users who post only a few tweets retweeted many times.

**Mentioning tweet rate, times mentioned, and conversation tweet rate:** Users with high mentioning tweet rate will be outgoing. The number of times the authorized users are mentioned will be large. The lonely users have small number of times mentioned. Conversation tweet rate indicates proportion of mentioning tweets in conversation, in other words, proportion of the following tweets: (1) tweets replying to mentioned tweets, and (2) tweets mentioning others which are replied to later. The interactive users will have high conversation tweet rate.

**Follower count, and follower-friend ratio:** The popular users have many followers, while the isolated users have small number of followers. In order to judge whether each user is mutually-connected, we use follower-friend ratio instead of matching a friend list and a follower list of the user due to the Twitter API rate limit.

Each Twitter user is assigned some of the user classes described in section 3. Some users may be assigned multiple classes, while some other users may be assigned no class.

## 5. Preliminary Experiment And Method for Finding Good Users to Follow

We carried out a preliminary experiment to estimate which user classes will be efficient for finding good users to follow.

### 5.1. Data Collection And Conditions for User Classification

Final goal of this study is improving accuracy of finding good users to follow. To achieve the goal, we pick up some users by the TURKEYS method [1] as pre-processing, then apply a newly proposed method to the user list. The pre-processing goes as follows.

1. Given an input query representing a topic of interest, get tweets matching the query posted in 5 days.
2. Remove duplicate tweets (the same tweet content posted by the same user).
3. Apply the TURKEYS method to the obtained tweets and get the top-20 users.

After the pre-processing, we collect the following data for each of the top-20 users to extract the features for the user classification.

1. The number of followers and friends of the user.
2. Recent 200 tweets in the user timeline.
3. Tweets mentioning the user.
4. Conversation tweets of the user's tweets mentioning others.

**Table 1.** Thresholds for User Classification

Active	Avg. tweet interval < 1 hour
Silent	Avg. tweet interval > 24 hours
Regular	CV of tweet interval < $\frac{1}{0.7}$ and avg. tweet interval < 24 hours
Occasional	CV of tweet interval > $\frac{1}{0.3}$ and avg. tweet interval < 24 hours
Repetitive	Duplicate tweet rate (as is, excl. entities, or URLs) > 0.3 and # tweets excl. retweets $\geq 50$
Random	Duplicate tweet rate (as is, excl. entities, or URLs) < 0.1 and # tweets excl. retweets $\geq 50$
Informative	(Avg. tweet length (as is) > 90 or avg. tweet length (excl. entities) > 70) and # tweets excl. retweets $\geq 50$
Chatting	(Avg. tweet length (as is) < 50 or avg. tweet length (excl. entities) < 30) and # tweets excl. retweets $\geq 50$
Attracting	Times retweeted > 50, # retweeted tweets > 10, or H-index of times retweeted > 5
Ignored	Times retweeted < 5, # retweeted tweets > 5, or H-index of times retweeted < 2
Outgoing	Mentioning tweet rate > 0.3 and # tweets excl. retweets > 50
Authorized	times mentioned > 50
Interactive	Conversation tweet rate > 0.5 and # mentioning tweets > 5
Lonely	times mentioned < 3
Popular	# followers > 1,000 and follower-friend ratio > 2.0
Mutually-connected	50 < # followers < 10,000 and 0.9 < follower-friend ratio < 1.1
Isolated	# followers < 10

We selected the following 7 Japanese keywords (in Japanese characters) as input query of the pre-processing: “nuclear power”, “animal test”, “whaling”, “dementia”, “ebook (digital book)”, “basic income”, and “fair trade”. For each keyword, we judged relevance between each of the top-20 users ranked by the pre-processing and the keyword on a scale of 0 to 2 in the same way as the judgment in [1].

Table 1 shows conditions (thresholds) for the user classification <sup>1</sup>. Multiple classes may be assigned to some users, while no class may be assigned to some other users (such users did not exist in this experiment).

## 5.2. Experimental Result

To estimate which user classes are efficient for finding good users to follow, we measured precision, recall, false discovery rate (FDR), and false positive rate (FPR) of each user classification.

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad \text{FDR} = \frac{FP}{TP + FP}, \quad \text{FPR} = \frac{FP}{TN + FP}$$

TP, FP, TN, and FN indicate the numbers of true positives, false positives, true negatives, and false negatives respectively. In this experiment, we consider both the relevance score of 1 and 2 as “positive”. If FDR or FPR is large, the classification would be efficient as a negative feature. Precision and FDR are more important than recall and FPR since false positives should be excluded in this task. The result is shown in Figure 1. A number

<sup>1</sup>This is for Japanese tweets. Setting will be different for different languages.

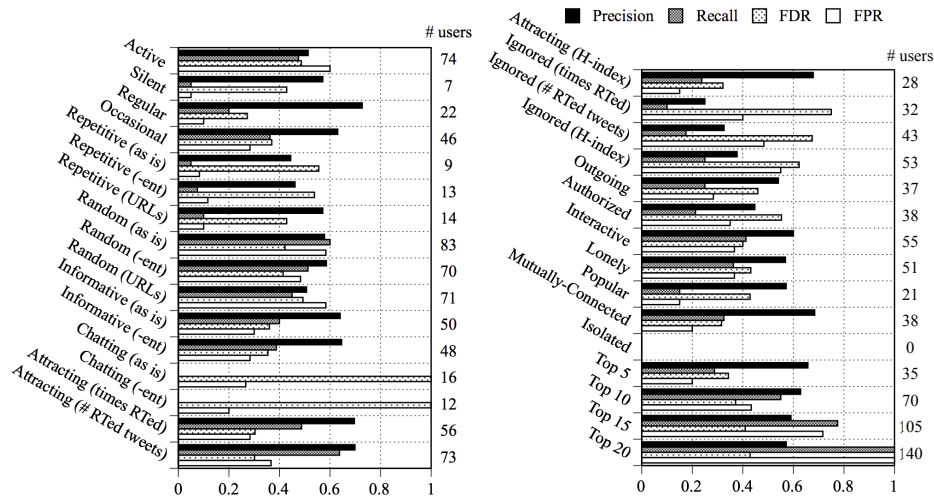


Figure 1. Precision, Recall, FDR, and FPR of User Classification

at the right of each class indicates the number of users assigned to the class. “Top- $N$ ” ( $N = 5, 10, 15, 20$ ) indicates the performance in the case that the top- $N$  users of each keyword are simply selected after the pre-processing. They are listed for comparison.

From the result, we can see that “regular”, “attracting”, and “mutually-connected” will be efficient from the viewpoint of precision (their precision is higher than precision of “Top 5”). “Chatting” and “ignored” will also be efficient as negative features since their FDR is high. Contrary to our expectation, FDR of “repetitive” is not high. There are two possible reasons. One reason is that some (excessively) repetitive users are already excluded in the pre-processing by removing duplicate tweets. The other reason is that some users sometimes post duplicate tweets notifying events held in the future, products they sell, and so on although they normally post tweets valuable for others. However, we think this is still efficient for excluding false positives.

### 5.3. Method for Finding Good Users to Follow

We judge whether each user is a good user to follow or not as shown in Figure 2 (“OK” and “NG” respectively mean the user is/is not a good user to follow). The chatting users are not good users to follow since they are usually post short tweets which do not have much information. The informative but ignored users are not good users to follow either since they may be spammers posting advertising tweets, which tend to be long. In the case that the target user is not ignored but random, the popular users or the mutually-connected users are good users to follow.

After the pre-processing, the top-20 users ranked by the pre-processing are reranked so that users judged positive by this method are ranked higher than users judged negative. Rank relation among users judged positive is kept, and rank relation among users judged negative is also kept.

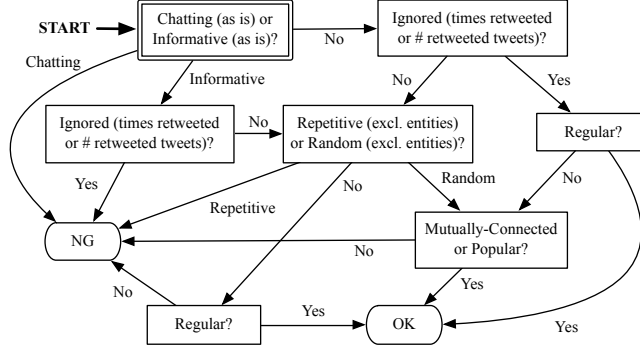


Figure 2. Flowchart of Finding Good Users to Follow

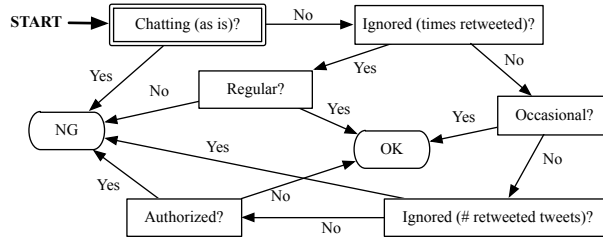


Figure 3. Simplified Decision Tree Produced by C4.5

## 6. Evaluation

We evaluate reranking of the top-20 users by our method with respect to normalized discounted cumulative gain (nDCG) [8] defined as follows.

$$DCG_N = rel_1 + \sum_{i=2}^N \frac{rel_i}{\log_2 i}, \quad nDCG_N = \frac{DCG_N}{\max DCG_N}$$

where  $rel_i$  is relevance score assigned to the  $i$ -th user and  $\max DCG_N$  is the DCG score of the top- $N$  users in the case that the 20 users are ranked in the ideal order (descending order of relevance score). For comparison, we produced a simplified decision tree by C4.5 [9] as shown in Figure 3. All of the 140 users are used as training data.

Result is shown in Table 2. Our method improves ranking result of the pre-processing in most cases. Compared with C4.5, nDCG score is almost equivalent on average although which method is better differs according to keyword.

Simplified decision tree produced by C4.5 (Figure 3) does not follow our intuition. For example, judgement of the ignored users with respect to times retweeted is done before judgement of the occasional users, while judgement of the ignored users with respect to the number of retweeted tweets is done after the judgement of the occasional users. It is also strange that the authorized users are judged negative while the non-authorized users are judged as positive. This decision tree might overfit the training data and our method is more generalized. Open evaluation (evaluation on unseen data) needs to be done to see which method is more generalized, which is left for future work.

**Table 2.** nDCG Score of Each Method

	Top 5	Top 10	Top 15	Top 20	Top 5	Top 10	Top 15	Top 20
Keyword	nuclear power				animal test			
Our method	<b>0.711</b>	<b>0.738</b>	<b>0.769</b>	<b>0.864</b>	<b>1.000</b>	<b>0.968</b>	0.929	0.986
C4.5	0.518	0.660	0.677	0.777	<b>1.000</b>	0.966	<b>0.949</b>	<b>0.987</b>
Pre-processing	0.472	0.627	0.677	0.776	<b>1.000</b>	0.878	0.900	0.976
Keyword	whaling				dementia			
Our method	<b>0.682</b>	0.772	0.772	0.834	0.940	<b>0.921</b>	0.949	0.973
C4.5	<b>0.682</b>	<b>0.880</b>	<b>0.880</b>	<b>0.880</b>	<b>1.000</b>	0.920	<b>0.952</b>	<b>0.976</b>
Pre-processing	0.598	0.696	0.696	0.758	<b>1.000</b>	0.849	0.909	0.957
Keyword	ebook				basic income			
Our method	<b>0.739</b>	<b>0.841</b>	<b>0.841</b>	<b>0.899</b>	<b>0.771</b>	<b>0.737</b>	0.767	<b>0.860</b>
C4.5	0.542	0.541	0.710	0.710	0.719	0.735	<b>0.789</b>	0.836
Pre-processing	0.402	0.423	0.485	0.631	0.542	0.609	0.701	0.770
Keyword	fair trade				Average			
Our method	0.177	0.430	0.498	0.555	0.717	0.772	0.789	<b>0.853</b>
C4.5	<b>0.579</b>	<b>0.756</b>	<b>0.756</b>	<b>0.756</b>	<b>0.720</b>	<b>0.780</b>	<b>0.816</b>	0.846
Pre-processing	0.070	0.321	0.459	0.516	0.583	0.629	0.689	0.769

## 7. Conclusion

In this paper, we presented a method for finding good Twitter users to follow based on user classification. The method is incorporated with our previous work, and experimental result showed our method improves the performance. The result is almost equivalent to result of a decision tree produced by C4.5, but the decision tree does not follow our intuition. In the future, we need to conduct open evaluation for detail analysis.

## References

- [1] Tomoya Noro, Fei Ru, Feng Xiao, and Takehiro Tokuda. Twitter user rank using keyword search. In *22nd European-Japanese Conference on Information Modelling and Knowledge Bases*, pages 48–65, 2012.
- [2] Kristian Slabbekoorn, Tomoya Noro, and Takehiro Tokuda. Towards Twitter user recommendation based on user relations and taxonomical analysis. In *23rd European-Japanese Conference on Information Modelling and Knowledge Bases*, pages 123–140, 2013.
- [3] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Who is tweeting on Twitter: Human, bot, or cyborg? In *26th Annual Computer Security Applications Conference*, pages 21–30, 2010.
- [4] Meeyoung Cha, Hamed Haddadi, Fabrício Benevenuto, and Krishna P. Gummadi. Measuring user influence in Twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media*, pages 10–17, 2010.
- [5] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the Web. In *9th International World Wide Web Conference*, pages 309–320, 2000.
- [6] Graham Neubig and Kevin Duh. How much is said in a tweet? A multilingual, information-theoretic perspective. In *AAAI 2013 Spring Symposium on Analyzing Microtext*, pages 32–39, 2013.
- [7] Jorge E. Hirsch. An index to quantify an individual’s scientific research output. *National Academy of Science of the United States of America*, 102(46):16569–16572, 2005.
- [8] Kalervo Jarvelin and Jaana Kekalainen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.
- [9] John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.



# An Axiomatic Approach to the Relational Concepts

Jari Palomäki

*Tampere University of Technology/Pori*  
*Pohjoisranta, P.O.Box 300, FIN-28101 Pori, Finland*  
 jari.palomaki@tut.fi

**Abstract.** In this paper an axiomatic approach to the relational concept theory, denoted by *RKC*, is presented. This axiomatic approach is based on the intensional containment relation between the under relations of the given relational concept. Also, it is proposed that an algebraic model for *RKC* is a *complete semilattice*, where every relational concept as a principal ideal generated by it defines a Boolean algebra.

**Keywords.** Concept, relation, intensional, model.

## Introduction

Gottfried Wilhelm Leibniz (1646-1716) worked out a variety of both intensional and extensional treatments of the logic of concepts. Raili Kauppi (1920-1995), influenced by the intensional aspect of Leibniz's logic, developed an axiomatic intensional concept theory in [1]. This axiomatic intensional concept theory, denoted by *KC*, is presented in a first-order language  $L$  that contains individual variables  $a, b, c, \dots$ , which range over the (*monadic*) *concepts*, and one non-logical 2-place *intensional containment relation*, denoted by " $\geq$ ". When  $a \geq b$ , we say that a concept  $b$  is intensionally contained in a concept  $a$ . This theory *KC* is further studied in [2], where it is proposed the model of it being a *complete semilattice*.

Kauppi advanced her concept theory also towards relational concepts, but that work remained unfinished. In this paper Kauppi's axiomatic intensional concept theory *KC* is extended to concern the relational concepts as well, denoted by *RKC*, and an axiomatic approach to relational concepts is presented.<sup>1</sup>

## 1. An Axiomatic Intensional Concept Theory *RKC*

An *intensional relational concept theory*, denoted by *RKC*, is presented in a first-order language  $L$  that contains variables  $p, r, s, \dots, x, y, z, \dots$ , which range over the *relational concepts*, and one non-logical 2-place *intensional containment relation*, denoted by " $\geq$ ". When  $r \geq s$ , we say that a relational concept  $r$  contains intensionally a relational concept  $s$ , or that the relational concept  $s$  is intensionally contained in the relational concept  $r$ .

Let us denote an  $n$ -place relational concept by  $r_n$ , where  $1 \leq n$ , and  $n \in \mathbb{N}$ .<sup>2</sup> A  $k$ -place under relation of  $r_n$  is denoted by  $U(\mu_1, \mu_2, \dots, \mu_k)r_n$ , where  $\langle \mu_1, \mu_2, \dots, \mu_k \rangle$  is a  $k$ -tuple of concepts, ( $1 \leq k \leq n$ ). The  $n$ -place

<sup>1</sup> Hannu Kangassalo has proposed an approach to conceptual modelling based on the above used intensional containment, for example, in [3] and also in [4].

<sup>2</sup> When  $n = 1$ , we have monadic concepts as studied in *KC*. When  $n = 0$ , we will have thoughts, or *Gedanke* (Frege), which are the intensions – the *senses* – of propositions. Thoughts, however, are not studied in this paper, but the similar treatment would also hold to them.

relational concept  $r_n$  can be *restricted* to a  $k$ -place under relation  $s_k$ , which will be denoted as  $r_n \upharpoonright [\mu_1, \mu_2, \dots, \mu_k : s_k]$  or shorter as  $r_n \upharpoonright s_k$ . The restriction is useful in practice, and gives the general idea of under relations.<sup>3</sup> However, it is not used in this axiomatic approach to the relational concepts developed as follows.

The first axiom of *RKC* states that if there is an under relation  $s$  of  $r$ ,<sup>4</sup> then the under relation  $s$  is intensionally contained in the relational concept  $r$ .

$$\text{AX}_{\text{Rel}} \quad (\exists s) s = \text{U}(\mu_1, \mu_2, \dots, \mu_k)r \rightarrow r \geq s.$$

A relation  $s$  is an *under relation* of the relational concept  $r$ , denoted by  $s \text{ U } r$ , if and only if there is a  $k$ -place under relation  $p$  of  $r$  and  $p$  contains intensionally  $s$ .

$$\text{Df}_{\text{U}} \quad s \text{ U } r =_{\text{df}} (\exists p) (p = \text{U}(\mu_1, \mu_2, \dots, \mu_k)r \wedge p \geq s).$$

Two relational concepts  $s$  and  $r$  are said to be *comparable*, denoted by  $s \text{ H}_{\text{U}} r$ , if there exists a relational concept  $x$  which is an under relation of them both.

$$\text{Df}_{\text{H}_{\text{U}}} \quad s \text{ H}_{\text{U}} r =_{\text{df}} (\exists x) (x \text{ U } s \wedge x \text{ U } r).$$

If two relational concepts  $s$  and  $r$  are not comparable, they are *incomparable*, which is denoted by  $s \text{ I}_{\text{U}} r$ .

$$\text{Df}_{\text{I}_{\text{U}}} \quad s \text{ I}_{\text{U}} r =_{\text{df}} \sim s \text{ H}_{\text{U}} r.$$

Dually, two relational concepts  $s$  and  $r$  are said to be *compatible*, denoted by  $s \perp_{\text{U}} r$ , if there exists a relational concept  $x$  which has both  $s$  and  $r$  of its under relations.

$$\text{Df}_{\perp_{\text{U}}} \quad s \perp_{\text{U}} r =_{\text{df}} (\exists x) (s \text{ U } x \wedge r \text{ U } x).$$

If two relational concepts  $s$  and  $r$  are not compatible, they are *incompatible*, which is denoted by  $a \text{ Y}_{\text{U}} b$ .

$$\text{Df}_{\text{Y}_{\text{U}}} \quad s \text{ Y}_{\text{U}} r =_{\text{df}} \sim s \perp_{\text{U}} r.$$

The next two axioms of *RKC* states that the under relation is a *reflexive* and *transitive* relation.<sup>5</sup>

$$\text{AX}_{\text{RefU}} \quad r \text{ U } r.$$

$$\text{AX}_{\text{TransU}} \quad p \text{ U } s \wedge s \text{ U } r \rightarrow p \text{ U } r.$$

Two relational concepts  $s$  and  $r$  are said to be *intensionally identical*, denoted by  $s \approx_{\text{U}} r$ , if the relational concept  $s$  is an under relation of the relational concept  $r$  and the relational concept  $r$  is an under relation of the relational concept  $s$ .<sup>6</sup>

$$\text{Df}_{\approx_{\text{U}}} \quad s \approx_{\text{U}} r =_{\text{df}} s \text{ U } r \wedge r \text{ U } s.$$

<sup>3</sup> As an example, let us have a three place relational concept  $give_3$ , which will be read as “ $x$  gives  $y$  to  $z$ ”. This relational concept has a 2-place under relation  $get_2$ , which we can read as “ $x$  get  $z$ ”, i.e.,  $give_3 \upharpoonright get_2$ . Thus, for example, the relational concept  $give_3$  between the concepts of a *mother*, a *ball*, and a *child*, i.e., “A mother gives a ball to the child”, has a 2-place under relation  $get_2$  between the concepts of the *child* and the *ball*, i.e.,  $give_3 \upharpoonright [child, ball : get_2]$ . Moreover, it has also a 3-place under relation  $get_3$ : as follows:  $give_3 \upharpoonright [child, ball, mother : get_3]$ , which is to be read: “A child get a ball from the mother”. As we can see, in these examples the word “get” means three different concepts, i.e.,  $get_2$ ,  $get_3$ , and  $get_3$ . In order to make the difference between  $get_3$  and  $get_3$  visible, we can put the order of placeness as subindices, i.e.,  $get_3 = get_{1,2,3}$  and so  $get_3 = get_{3,2,1}$ , and thus  $get_2 = get_{3,2}$ . Generally, an  $n$ -place relational concept has  $\sum_{k=1}^{k=n} P(n, k)$  different  $k$ -place under relations, where  $P(n, k) = n!/(n-k)!$ , and  $n, k \in \mathbb{N}$ .

<sup>4</sup> In what follows the number of the placeness of the relational concepts indicated in subscripts will be omitted, when no confusions results.

<sup>5</sup> That is, the under relation is a *pre-ordering* on a set  $R$  of relational concepts.

<sup>6</sup> An under relation can be defined to be *antisymmetric* only on the quotient set  $R/\approx_{\text{U}} =_{\text{df}} \{[r] \mid r \in R\}$ , where  $R$  is the set of relational concepts,  $[r]$  is the equivalence class of relational concepts  $x$  modulo the intensional identity relation  $\approx_{\text{U}}$ , i.e.  $[r] =_{\text{df}} \{x \mid r \approx_{\text{U}} x\}$ . Thus, on the quotient set  $\text{RC}/\approx_{\text{U}}$  we will get a *partial order* among relational concepts by defining  $[s] \text{ U } [r] \leftrightarrow s \text{ U } r$ .

The intensional identity between relational concepts is clearly a reflexive, symmetric and transitive relation, hence an equivalence relation.

A relational concept  $p$  is called an *intensional product* of two relational concepts  $s$  and  $r$ , if any relational concept  $x$  is an under relation of  $p$  if and only if it is an under relation of both  $s$  and  $r$ . If two relational concepts  $s$  and  $r$  have an intensional product, it is unique up to the intensional identity and we denote it then by  $s \otimes_U r$ .

$$\text{Df}_{\otimes_U} \quad p \approx_U s \otimes_U r \text{ =_{df} } (\forall x) (x \text{ U } p \leftrightarrow x \text{ U } s \wedge x \text{ U } r).$$

The following axiom  $\text{Ax}_{\otimes_U}$  of *RKC* states that if two relational concepts  $s$  and  $r$  are comparable, there exists a relational concept  $p$ , which is their intensional product.

$$\text{Ax}_{\otimes_U} \quad s \text{ H}_U r \rightarrow (\exists p) (p \approx_U s \otimes_U r).$$

It is easy to show that the intensional product is idempotent, commutative, and associative.

A relational concept  $p$  is called an *intensional sum* of two relational concepts  $s$  and  $r$ , if the relational concept  $p$  is an under relation of any relational concept  $x$  if and only if both  $s$  and  $r$  are under relations of it.

If two relational concepts  $s$  and  $r$  have an intensional sum, it is unique up to the intensional identity and we denote it then by  $s \oplus_U r$ .

$$\text{Df}_{\oplus_U} \quad p \approx_U s \oplus_U r \text{ =_{df} } (\forall x) (p \text{ U } x \leftrightarrow s \text{ U } x \wedge r \text{ U } x).^7$$

The following axiom  $\text{Ax}_{\oplus_U}$  of *RKC* states that if two relational concepts  $s$  and  $r$  are compatible, there exists a relational concept  $p$ , which is their intensional sum.

$$\text{Ax}_{\oplus_U} \quad s \perp_U r \rightarrow (\exists p) (p \approx_U s \oplus_U r).$$

The intensional sum is idempotent, commutative, and associative.

The intensional product of two relational concepts  $s$  and  $r$  is an under relation of their intensional sum whenever both sides are defined.

$$\text{Th}_U 1 \quad (s \otimes_U r) \text{ U } (s \oplus_U r).$$

*Proof:* If  $s \otimes_U r$  exists, then by  $\text{Df}_{\otimes_U}$ ,  $(s \otimes_U r) \text{ U } s$  and  $(s \otimes_U r) \text{ U } r$ . Similarly, if  $s \oplus_U r$  exists, then by  $\text{Df}_{\oplus_U}$ ,  $s \text{ U } (s \oplus_U r)$  and  $r \text{ U } (s \oplus_U r)$ . Hence, by  $\text{Ax}_{\text{TransU}}$ , the theorem follows.

The next axiom of *RKC* concerns the distributivity of an intensional sum and a product whenever both sides are defined.

$$\text{Ax}_{\text{DistrU}} \quad (s \otimes_U (r \oplus_U p)) \text{ U } ((s \otimes_U r) \oplus_U (s \otimes_U p)).^8$$

A relational concept  $r$  is an *intensional negation* of a relational concept  $s$ , denoted by  $\neg s$ , if and only if it is an under relation of all those relational concepts  $x$ , which are intensionally incompatible with the relational concept  $s$ . When  $\neg s$  exists, it is unique up to the intensional identity.

$$\text{Df}_{\neg_U} \quad r \approx_U \neg s \text{ =_{df} } (\forall x) (r \text{ U } x \leftrightarrow x \text{ Y}_U s).$$

<sup>7</sup> Thus,  $s \otimes_U r \leftrightarrow [s] \otimes_U [r]$  is a greatest lower bound in  $R/\approx_U$ , whereas  $s \oplus_U r \leftrightarrow [s] \oplus_U [r]$  is a least upper bound in  $R/\approx_U$ .

<sup>8</sup> Since  $((s \otimes_U r) \oplus_U (s \otimes_U p)) \text{ U } (s \otimes_U (r \oplus_U p))$  holds always whenever both sides are defined, we get by  $\text{Ax}_{\text{DistrU}}$  the following intensional identity,  $(s \otimes_U (r \oplus_U p)) \approx_U ((s \otimes_U r) \oplus_U (s \otimes_U p))$ , which implies the dual,  $(s \oplus_U (r \otimes_U p)) \approx_U ((s \oplus_U r) \otimes_U (s \oplus_U p))$ .

The following axiom  $Ax_{\neg U}$  of *RKC* states that if there is a relational concept  $x$  which is incompatible with the relational concept  $r$ , then there exists a relational concept  $y$ , which is the intensional negation of the relational concept  $r$ .

$$Ax_{\neg U} \quad (\exists x) (x \mathop{Y}_U r) \rightarrow (\exists y) (y \approx_U \neg r).$$

It can be proved that a relational concept  $r$  contains intensionally its intensional double negation provided that it exists.

$$Th_U 2 \quad \neg \neg r \mathop{U} r.^9$$

*Proof:* By  $Df_{\neg U}$  the equivalence (1):  $\neg r \mathop{U} s \leftrightarrow s \mathop{Y}_U r$  holds. By substituting  $\neg r$  for  $s$  to (1), we get  $\neg r \mathop{U} \neg r \leftrightarrow \neg r \mathop{Y}_U r$ , and so, by  $Ax_{RefU}$ , we get (2):  $\neg r \mathop{Y}_U r$ . Then, by substituting  $r$  for  $s$  and  $\neg r$  for  $r$  to (1), we get  $\neg \neg r \mathop{U} r \leftrightarrow r \mathop{Y}_U \neg r$  and hence, by (2), the theorem follows.

Also, the following forms of the *De Morgan's formulas* can be proved whenever both sides are defined:<sup>10</sup>

$$Th_U 3 \quad \begin{array}{l} \text{a) } \neg(s \oplus r) \mathop{U} \neg s \otimes \neg r, \\ \text{b) } \neg(s \otimes r) \approx_U \neg s \oplus \neg r. \end{array}$$

*Proof:* First we are to proof the following important lemma:

$$Lemma_U 1 \quad r \mathop{U} s \rightarrow \neg s \mathop{U} \neg r.$$

*Proof:* From  $r \mathop{U} s$  follows  $(\forall x) (x \mathop{Y}_U r \rightarrow x \mathop{Y}_U s)$ , and thus by  $Df_{\neg U}$  the  $Lemma_U 1$  follows.<sup>11</sup>

i) If  $s \oplus_U r$  exists, then by  $Df_{\oplus U}$ ,  $s \mathop{U} s \oplus_U r$  and  $r \mathop{U} s \oplus_U r$ . By  $Lemma_U 1$  we get  $\neg(s \oplus_U r) \mathop{U} \neg s$  and  $\neg(s \oplus_U r) \mathop{U} \neg r$ . Then, by  $Df_{\otimes U}$ ,  $Th_U 3a$  follows.

ii) This is proved in the four steps as follows:

$$1. \neg s \oplus_U \neg r \mathop{U} \neg(s \otimes_U r).$$

Since  $s \otimes_U r \mathop{U} s$ , it follows by  $Lemma_U 1$  that

$$\neg s \mathop{U} \neg(s \otimes_U r). \text{ Thus, by } Df_{\oplus U}, 1 \text{ holds.}$$

$$2. \neg(s \otimes_U r) \mathop{U} \neg(\neg \neg s \otimes_U \neg \neg r).$$

Since  $\neg \neg s \mathop{U} s$ , by  $Th_U 2$ , it follows by  $Df_{\otimes U}$  that

$$(\neg \neg a \otimes_U \neg \neg b) \mathop{U} (a \otimes_U b). \text{ Thus, by } Lemma_U 1, 2 \text{ holds.}$$

<sup>9</sup> This relation does not hold conversely without stating a further axiom  $Ax_{\neg U}$ :  $r \mathop{Y} \neg s \rightarrow s \mathop{U} r$ . Thus,  $r \mathop{U} \neg \neg r$ , and hence by  $Th_U 2$ ,  $r \approx_U \neg \neg r$  holds only, if the relational concept  $r$  is an under relation of the every concept  $s$ , which is incompatible with the intensional negation of the concept  $r$ .

<sup>10</sup> From this onwards it is presumed without explicit mentioning that the formulas hold whenever both sides of formulas are defined.

<sup>11</sup> Neither this lemma holds conversely without a further axiom  $Ax_{\neg U}$ . Note that the function from  $r$  to  $\neg r$  inverts order, and therefore it carries intensional sums to intensional products and intensional products to intensional sums provided, of course, that they exist.

$$3. \neg(\neg s \oplus_U \neg r) \cup (\neg\neg s \otimes_U \neg\neg r).$$

Since  $s \cup (s \oplus_U r)$ , it follows by Lemma<sub>U</sub> 1 that

$$\neg(s \oplus_U r) \cup \neg s, \text{ and so, by Df}_{\otimes_U}, \text{ it follows}$$

$\neg(s \oplus_U r) \cup (\neg s \otimes_U \neg r)$ . Thus, by substituting  $\neg s$  for  $s$  and  $\neg r$  for  $r$  to it, 3 holds.

$$4. \neg(s \otimes_U r) \cup \neg s \oplus_U \neg r.$$

Since  $\neg\neg(\neg s \oplus_U \neg r) \cup \neg s \oplus_U \neg r$ , by Th<sub>U</sub> 2, and from 3 it follows by Lemma<sub>U</sub> 1 that  $\neg(\neg\neg s \otimes_U \neg\neg r) \cup \neg\neg(\neg s \oplus_U \neg r)$ , and by Ax<sub>TransU</sub> we get,  $\neg(\neg\neg s \otimes_U \neg\neg r) \cup \neg s \oplus_U \neg r$ . Thus, by 2 and by Ax<sub>TransU</sub>, 4 holds.

From 1 and 4, by Df<sub>≈U</sub>, the Th<sub>U</sub> 3b follows.

If a relational concept  $r$  is an under relation of every concept  $x$ , the relational concept  $r$  is called a *general relational concept*, and it is denoted by  $GR$ . The general relational concept is unique up to the intensional identity, and it is defined as follows.

$$\text{Df}_{GRU} \quad r \approx_U GR =_{\text{df}} (\forall x) (r \cup x).^{12}$$

The next axiom of  $RKC$  states that there is a relational concept, which is intensionally contained in every relational concept.

$$\text{Ax}_{GRU} \quad (\exists x)(\forall y) (x \cup y).$$

Adopting the axiom of the general relational concept it follows that all relational concepts are comparable. Since the general relational concept is compatible with every relational concept, it has no intensional negation.

A *special relational concept* is a relational concept  $r$ , which is not intensionally contained in any other relational concept except for relational concepts intensionally identical to itself. Thus, there can be many special relational concepts.

$$\text{Df}_{SU} \quad S(r) =_{\text{df}} (\forall x) (r \cup x \rightarrow x \cup r).^{13}$$

The last axiom of  $RKC$  states that there is for any relational concept  $y$  a special relational concept  $x$ , which has  $y$  as its under relation.

$$\text{Ax}_{SU} \quad (\forall y)(\exists x) (S(x) \wedge y \cup x).$$

Since the special relational concept  $r$  is either compatible or incompatible with every relational concept, the *law of excluded middle* holds for  $r$  so that for any relational concept  $x$ , which has an intensional negation, either the relational concept  $x$  or its intensional negation  $\neg x$  is under relation of it. Hence, we have proved the following theorem.

$$\text{Th}_U 4 \quad (\forall x) S(r) \rightarrow (x \cup r \vee \neg x \cup r).$$

We may take Th<sub>U</sub> 4 to be as a kind of syntactic completeness theorem of  $RKC$ .<sup>14</sup>

<sup>12</sup> So,  $RG \leftrightarrow [RG]$  is a least element in  $R/\approx_U$ .

<sup>13</sup> That is,  $S(r) \leftrightarrow [r]$  is a maximal element in  $R/\approx_U$ .

<sup>14</sup> A special concept in  $KC$  corresponds Leibniz's *complete concept of an individual*, which means that a particular individual would contain one member of every pair of mutually incompatible concepts. Leibniz studied mostly monadic concepts.

## 2. An Algebraic Model of *RKC*

The axioms of *RKC* are analogous to the axioms of *KC*, cf. [2]. There is only one additional axiom,  $Ax_{Rel}$ , which is a special axiom for the relational concepts, since the monadic concepts have only themselves as their under relations, which is trivial. We can say that *KC* is a *sub-theory* of *RKC*, or that *RKC* is an *extension* of *KC*.<sup>15</sup>

The Completeness Theorem says that every consistent first-order theory has a model. In [2] it was proposed that an algebraic model of *KC* is a *complete semilattice*, where every concept  $a \in C$  defines a *Boolean algebra*  $B_a = \langle \downarrow a, \otimes, \oplus, \neg, G, a \rangle$ , where  $\downarrow a$  is an ideal, known as the *principal ideal generated by a*, i.e.,  $\downarrow a =_{\text{def}} \{x \in C \mid a \geq x\}$ , and the intensional negation of a concept  $b \in \downarrow a$  is interpreted as a *relative complement* of  $a$ . Based on that result, and that the *RKC* is an extension of *KC*, we can propose that an algebraic model of *RKC* is also a *complete semilattice*, where every relational concept  $r \in R$  defines a *Boolean algebra*  $B_r = \langle \downarrow r, \otimes_U, \oplus_U, \neg, GR, r \rangle$ . Thus,  $B_r$  is an *elementary extension* of  $B_a$ , or, in other words, that  $B_a$  is an *elementary substructure* (or *elementary submodel*) of  $B_r$ .<sup>16</sup>

## References

- [1] Kauppi, R.: *Einführung in die Theorie der Begriffssysteme*. Acta Universitatis Tampereensis. Ser. A. Vol. 15. Tampereen yliopisto, Tampere (1967).
- [2] Palomäki, J.: *From Concepts to Concept Theory: Discoveries, Connections, and Results*. Acta Universitatis Tampereensis. Ser. A, vol. 416. Tampereen yliopisto, Tampere (1994).
- [3] Kangassalo, H.: "COMIC: A system and methodology for conceptual modelling and information construction." *Data and Knowledge Engineering* **9**, 287-319 (1992/93).
- [4] Kangassalo, H.: "Approaches to the Active Conceptual Modelling of Learning." Eds. Chen, P.P. and Wong, L.Y.. *Active Conceptual Modeling of Learning: Next Generation Learning-Base System Development*. LNCS vol. 4512, pp. 168-193. Springer, Heidelberg (2007).
- [5] Palomäki, J. and Kangassalo, H.: "That Is-IN Isn't Is-A: A Further Analysis of Taxonomic Links in Conceptual Models". *Advances in Knowledge Representation*. Ed. C. Ramirez. InTech: Rijeka, (2012), 3-18. Available from: <http://www.intechopen.com/books/advances-in-knowledge-representation/that-is-in-isn-t-is-a-a-further-analysis-of-taxonomic-links-in-conceptual-modelling>
- [6] Imaguire, G.: "Logic and Intensionality". *Principia* **14**, (2010), 111-124.

<sup>15</sup> The concept of extension is now used in a syntactic level, and *not* as an extension of a concept.

<sup>16</sup> Note that the models of *KC* and *RKC* are models for conceptual structures. Accordingly, relational concepts are possible to understand being concepts in the similar way as monadic concepts are concepts, except that their intensions determines both i) how many place relations they are and ii) how many under relations they have. The extensions of monadic concepts are sets, or classes, whereas the extensions of  $n$ -place relational concepts are sets of ordered  $n$ -tuples. The relationships between extensions and intensions are further studied in [2] and [5] – and that extensions do not determine intensions is also realized and convincingly argued for in [6].

## *Challenge in Urban Flood Mitigating System: Decision Support based on Cyber-Physical-Human Infrastructure*

*Dadet Pramadihanto<sup>1)</sup>, Wahyu T Sesulihatien<sup>1,2)</sup>, Soffi Patrisia<sup>1)</sup>, Shiori Sasaki<sup>2)</sup>, Yasushi Kiyoki<sup>2)</sup>*

*<sup>1)</sup> Electronics Engineering Polytechnic Institute of Surabaya, Indonesia*

*<sup>2)</sup> Keio University, Shonan Fujisawa Campus, Japan*

**Abstract.** *Currently, floods are not only occurred in the outskirts of the river course, but also in the urban area, especially in the big city. The main problem of urban flood is the fact that it occurs in highly populated areas. It is a global phenomenon that causes widespread devastation, economic damages and loss of human lives. All the strategies basically are good for long term mitigation but not appropriate for solving the real problems when a disaster happens, because it is static and not real time. To overcome, two main points should be developed: socio-cultural knowledge on floods and flood prevention infrastructure development. Both are correlated to build the settlement of flood problem. Therefore, it is essential to build an integrated system combining Cyber-Physical-Human. The proposed system includes (1) physical layer that consist of sensors rainfall and river water levels and satellite sensors, (2) abstract layer consist of flood modelling (3) interaction with human. Mitigation system based Cyber - Physical - Human will be very useful for agencies related to flood control and as a decision making tool for the government and society at large. Surabaya is chosen as study area*

**Keywords.** *Cyber-Physical-Human, urban flood, flood-spread prediction, mitigation, sensor*

## Introduction

Indonesia is known as one of the vulnerable countries to flood disaster. Currently, floods are not only occurred in the outskirts of the river course, but also in the urban area, especially in the big city in Indonesia [1]. Flooding in urban areas is not only in the consequence of nature-made phenomenon such as heavy rainfall but also man-made event associated with their activities with lack of drainage [2]. The main problem of urban flooding is the fact that it is occurred in highly-populated areas. It is a global phenomenon that causes widespread devastation, economic damages and loss of human lives [3]. For this reason, an effective strategy in mitigation plays the important role.

The strategies in mitigation are different for every country. For example, in Ho Chi Minh, Vietnam, mitigation is focused on infrastructure due to socio economic planning [4], in Bangkok, Thailand, adaptation planning in climate change is chosen as solution [5], in Brisbane, Australia, adaptation strategies addressing flood risk management issues of an urban area with intensive residential and commercial uses [6]. In Indonesia, the strategy is emphasized on the infrastructure planning based on the history of floods happened in past [7]. All strategies basically are good for long term mitigation but not appropriate for solving the real problems when a disaster happens, because it is static and not real time. Some researches in several countries propose another real-time method based on the characteristics of their countries. In Bombay, real-time mitigation is implemented to maintain flow at pre-determined levels [8], and UK applied a system to predict the spatial and temporal distribution of both rainfall and surface flooding [9]. All methods are not comparable, because they are all unique in accordance with the cultural and geographic condition.

Flood mitigation involves not only planning but also community participation [10]. From this point of view, there are two main points should be developed: socio-cultural knowledge on floods and flood prevention infrastructure development. Until now, flood management in Surabaya is sporadic and unstructured [11]. Therefore, it is essential to build an integrated system that consist of a water sensor (rain, river streams, etc.), remote sensing by satellite (land use, DEM, etc.), drainage networks (river, drainage, dams), and people (human, government and policy makers), and cyber-infrastructure to a model and ingrate them. The integration is famous as Cyber-Physical-Human System.

This paper addresses to build prototype of Cyber Physical Human for mitigating urban flood in Surabaya. This system includes physical rainfall sensor flood modeling and pattern of drainage system as a picture of city characteristic. Output of the system is real time prediction of flood spreading as an early warning system. The main impact is an early response and evacuation by prediction of flood area. In long-term, it will built flood history map that can be used for flood prevention infrastructure drainage networks development planning.

### 1. An Approach on the CPH for Urban Flood

Cyber - Physical - Human (CPH) is a new research field that integrates cyber (virtual world), physical (sensor) and human (interaction) [12][13][14]. These systems are often implemented for public safety aspect, for example emergency disaster [13] and evacuation [14]. CPH system consists of three main components: the physical



elements to be controlled, cyber elements that represent communication links and software, and human social interaction as a representation relating to the physical elements that are controlled [12]. Framework of CPH is focused on the issue in which is constructed by the scenario [15]. CPH framework is prepared from element with refer to the environment; the main elements of the CPH is a sensor system, an abstract layer, and human scenario.

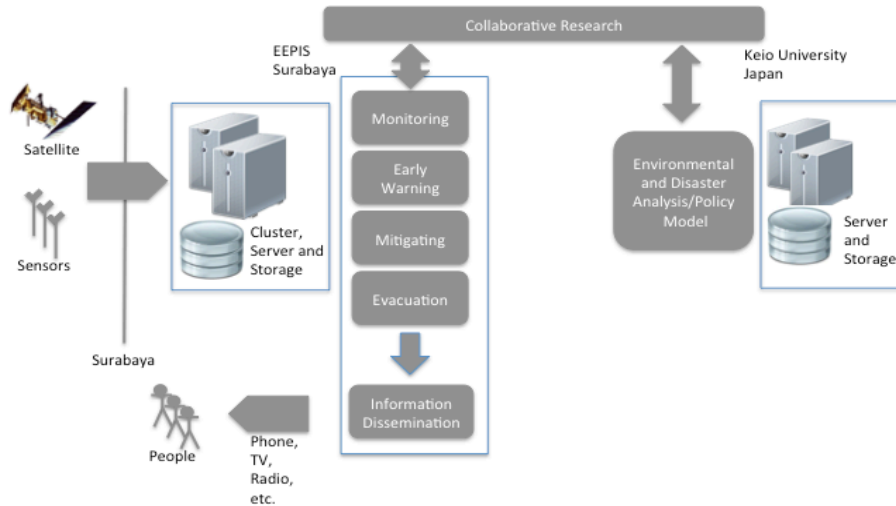
In this study, we focus on mitigation system of flooding in urban areas which characteristics is different with river flood mitigation. Two main aspect of mitigation in this study are processing data of sensors: environment and satellite, and a predicting flooded areas to prevent greater damage. The prediction data will be input for DSS evacuation and disaster response.

This system is expected to be useful for every different level. For example flood-related agencies can perform decision in real time condition, people will get information about the level of flood hazards in their respective regions, specific communities such as industrial or business can follow up the information to rescue their assets and Government can develop decision-making relating to the handling of the disaster and post- disaster recovery.

## **2. Systems Design and Case Study of Urban flood in Surabaya City**

Surabaya city is second biggest city in Indonesia. Located in east Java Island the most populated island in Indonesia. Geographically, Surabaya is close to Java Sea and most area is flat. The elevation ranges between 0 meter to 30 meters above the sea level. Heavy tropical rain normally hit Surabaya during the rainy season every year.

Urban flood management focuses on developing a cyber-infrastructure for urban flood management. Cyber structure will be employed for handling the data collection and integration, the data management, data mining and knowledge extraction. Methodological approach in line with the phases of disaster management: that is preparedness, mitigation, response, and recovery. In general, the system is illustrated in Figure 1.



**Figure 1.** Overall System

Figure 1 shows overall flood management system in Surabaya. System consists of monitoring, early warning, mitigation, evacuation system and information dissemination system. In this research we collaborate with Keio University to analyze impact of disaster and provide recommendation policy based on Keio University model. In this paper, we focus on flood spreading model for mitigation.

### 2.1. Abstract Layer: Mitigation Modeling

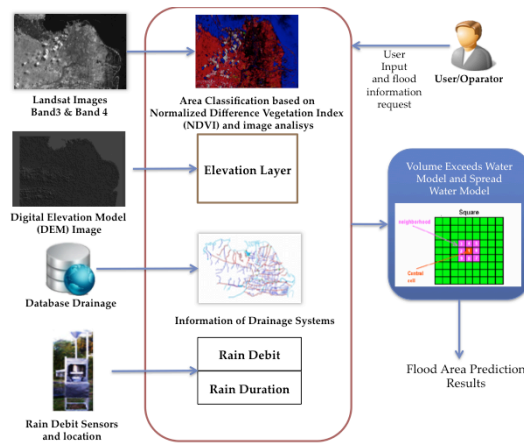
Disaster response in urban flood management generally relates to control, flood mitigation, evacuation, and disaster relief during floods. Mitigation will focus on the spread of a flood with the spatiotemporal analysis. System design is illustrated in figure 2. While the evacuation and disaster response focused on the evacuation of residents and optimization of alternative roads during flood period.

Figure 2 shows the system for modeling the spread of flood. This system integrates satellite images, land elevation, drainage network, flood histories and water level/rain sensors to perform flood spreading. The Landsat satellite images with 30m by 30m resolution per pixel is processed to get information about land elevation, land infiltration, and land classification (land, drainage/river, road, greenery). These parameters will be discussed in section 2.2. Another data about water level and rainfall are gathered from physical sensor. As a spreading method, we implemented 2-dimensional cellular automata lattice/ cell.

Algorithm of system is as follows:

- a. Rains pour in into the certain areas. As a nature of water, it will run to the lowest elevation cell. If the rainfall exceeds infiltration capacity of the land, the water overflows into the next lowest cell. This process will continuous until it reaches saturating condition.
- b. If the cell is part of a drainage channel or river, the water will fill the cell drainage pathways.
- c. From the results of (a), locations of overflow water can be determined and inundation area can be obtained by counting the number of flood cell.

- d. From the results of (a), depth of inundation flood can be predicted by calculating the difference in higher elevation cell and the lower one.
- e. In the other hand, if the filling water in drainage channel is over its capacity, the water will overflow. The flow of water will spread and predicted by following procedure (a).
- f. When the rain stopped, flood subsidence is calculated by procedure (a).



**Figure 2.** Flood spread modeling.

Mitigation scenario proposed in this paper is based on the pattern of the spread. Spread prediction consist of processing the sensor data of rainfall, topographic data from satellites, and classification of soil types. This prediction will apply on each lattice of cellular automata, a technique similar to the GIS-based urban flood inundation model (GUFIM) [16] to calculate the volume of water in each lattice.

$$R(t, x, y) = P(t, x, y) - F(t, x, y) - D(t, x, y) \quad (1)$$

where,  $R(t, x, y)$  is the excess rain water volume ( $m^3$ ),  $P(t, x, y)$  is the total volume of rainfall,  $F(t, x, y)$  is the total volume of rain water is absorbed, and  $D(t, x, y)$  is the total volume of the incoming rainwater drainage (rivers, water pump, etc.).

Relation between the absorption capacity of the soil and time is expressed in the following Horton equation:

$$F(t, x, y) = f_c + (f_0 - f_c)e^{-kt} \quad (2)$$

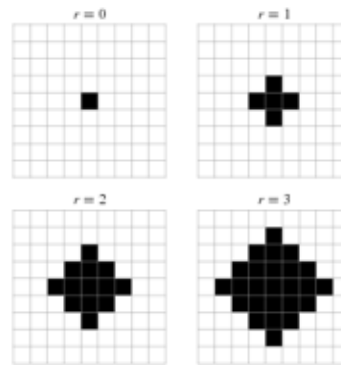
where,  $f_0$  is the infiltration capacity (soil absorption capacity) at any time,  $f_0$  is the initial infiltration capacity at  $t = 0$ ,  $f_c$  is the infiltration capacity after reaching a constant,  $k$  is a positive constant that depends on the soil and plant ground cover,  $t$  is the time.

Total volume entering the drainage is

$$D(t, x, y) = QtA \quad (3)$$

where,  $Q$  is the water discharge,  $t$  is time, and  $A$  is the cross-sectional area of drainage.

Excess rainwater  $R(t, x, y)$  is a dynamic depend on changes in influenced variables. But essentially the water tends to run in to a lower place. In case of e same elevation, the water will remain stagnant. Then, 2-dimension cellular automata method is implemented for predicting spreading based on mathematical models of flow in equation (2). Figure 3 shows the 2-dimensional lattice (cell) of cellular automata with von Neumann neighborhood [17].



**Figure 3.** Cellular Automata arrangement.

Neighborhood cells are cells that will worked on the flooding cell in cellular automata. Neighborhood of a cell generally consists of the cells in the vicinity. One model is the Von Neumann neighborhood, where the neighboring cells are shaped as a diamond. This scheme can be used to define a set of cells that surround a particular cell  $(x_0, y_0)$ , which can affect the development of two-dimensional cellular automata on a square grid. The radius of the Von Neumann neighborhood is defined by equation 2.3.

$$N_{(x_0, y_0)}^v = \{(x, y): |x - x_0| + |y - y_0| \leq r\} \quad (4)$$

Illustration Von Neumann neighborhood of radius,  $r = 0, 1, 2, 3$  shown in Figure 3. The number of cells in the Von Neumann neighborhood is with  $2r(r+1)+1$ .

## 2.2. Physical Layer

In this study, two sensors are employed: physical sensors and remote sensing. A remote sensing satellite data is operated to perform area characteristic due to flood and classify them, as well as the elevation of the land. Procedure of satellite data processing is shown in Figure 8.

Input is the raw data from the Landsat 5 TM satellite imagery (Thematic Mapper). This procedure is taken on as follows:

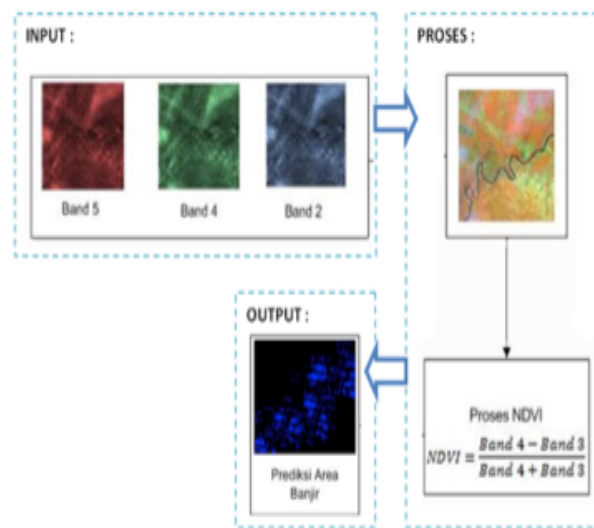
- Separating raw satellite image data based on the wavelength spectrum
- Constructing RGB image by combining each pixel of each band to RGB image
- Calculating NDVI, soil classification, and Cloud Reduction process.

- Alignment with GIS coordinates to integrate with GIS.

Image normalization or NDVI (Normalized Difference Vegetation Index) [18] is a calculation method to determine the image level of greenness, which is represent initial zoning of vegetation. NDVI can indicate parameters associated with vegetation such as biomass green foliage, green foliage area, the NDVI value that can estimate vegetation classification. In this data satellite, land use NDVI value is obtained by calculation of near infrared, and visible light reflected for vegetation. NDVI values are obtained by comparing the data reduction, near infrared and visible with the second summation data. The following calculation formula is used in Landsat satellite imagery:

$$NDVI = \frac{NIR-Visible}{NIR+Visible} \quad (5)$$

Where NIR is numeric processing result of channel 4 and visible is numeric-processing result of channel 3. Range of NDVI values is in between -1.0 to +1.0. Value that greater than 0.1 usually indicates increasing degrees of vegetation greenness and intensity. Value between 0 and 0.1 are generally representing characteristics of the rocks and vacant land. And a value less than 0 indicate the possibility of ice, clouds, water vapor cloud and snow. Surface vegetation NDVI value ranges from 0.1 to Savanna land (pasture) to 0.8 for tropical rain forest area.



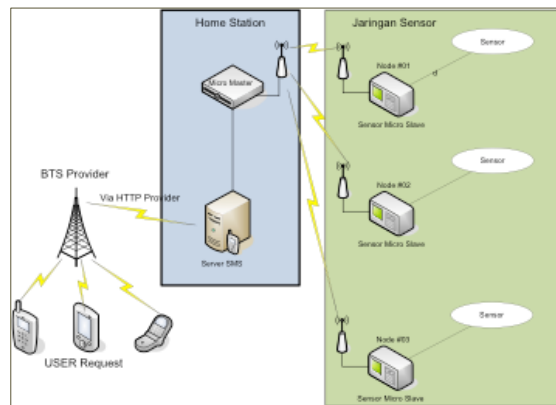
**Figure 4.** Block diagram of Satellite data processing

For Landsat 5 satellite image, data for calculating the NDVI is collected from channel 4 as a near-infrared channel data. So the formula is written as follows:

$$NDVI = \frac{Band\ 4 - Band\ 3}{Band\ 4 + Band\ 3} \quad (6)$$

From the results of equation (4) and (5), we also obtain data on land and water. In case of water, it is also required to compare with RGB values in order to minimize misclassification, because the NDVI value of water and the clouds are very similar.

For physical layer, we build the hardware system that consists of strain gauge sensors as rain sensors, and ultrasonic sensors as water level sensors. This sensor is integrated as wireless sensor. Block diagram wireless mesh sensor network is seen in Figure 5. Both rainfall sensor and water level sensor are placed on the main drainage channel and the river to monitor rainfall level and water level in the drainage channels and rivers. Each sensor read data and stores them on micro slave modules. Data communication among sensor is handled by wireless communication, and data communication from node to home station is handled by short message service whenever server request data, micro master will send it by Short Message Services



**Figure 5.** Sensors block diagram

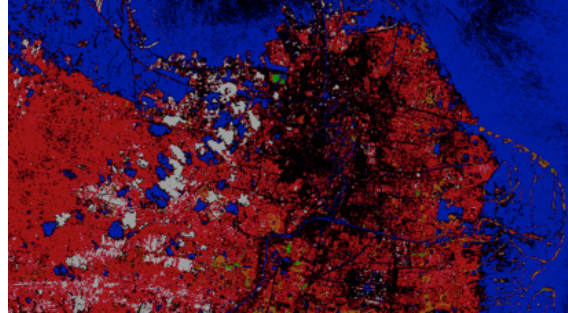
In this study, sensor data is placed in Pintu Air Jagir, located in Jagir River Surabaya. Rainfall data is measured every 10 minute but regarding with simulation, the highest rainfall data is chosen as input in simulation.

### 3. Results and Discussion

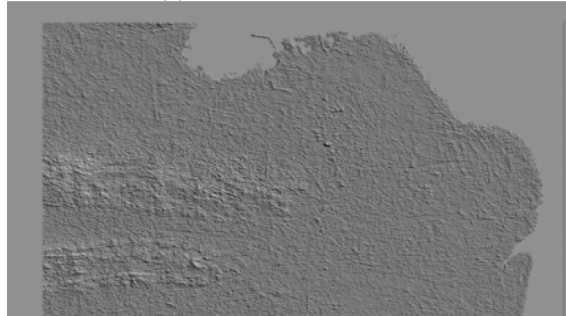
#### 3.1. Satellite Image Processing

Satellite image processing for urban flood consist of four procedures as illustrated in figure 6 (a)-(d).

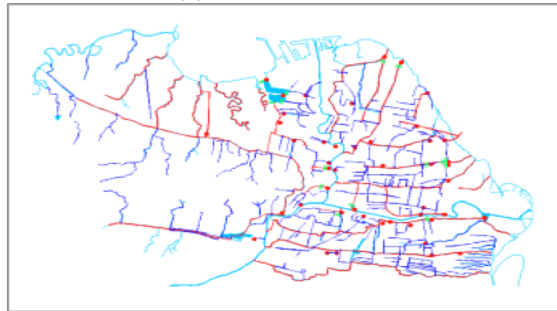
- Calculate and classify NDVI and RGB to get land use data
- Determine Digital Elevation Model (DEM) from scaling satellite data
- Mapping drainage system of Surabaya city
- Combining (a),(b) and (c)



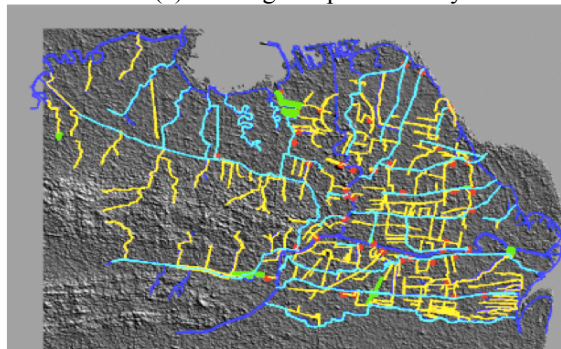
(a) Result of NDVI and RGB



(b) DEM of Satellite data



(c) Drainage map of Surabaya



(d) Combination of (a), (b) and (c)  
**Figure 6.** Satellite image processing.

The first procedure is NDVI calculation. It is purposed to determine the classification of an area based on a certain scale. Basically, from Landsat satellite data we put 3 channels:

- Band 5 (red): distinguishing types and condition of plants, distinguishing clouds, snow, ice
- Band 4 (green): investigating biomass plants and also differentiates land boundaries and land-plant-water.
- Band 3 (blue): detecting plant

NDVI values are calculated by equation (8). And classification results are shown in table 1.

**Table1.** Result of NDVI and RGB

<i>Classification Area</i>	<i>Range of NDVI and RGB</i>
<i>Water</i>	$NDVI = -0.3 - 0$ $R = 16 - 50, G = 18 - 59, B = 8 - 255$
<i>Cloud</i>	$NDVI = -1 - 0.1$ $R = 110 - 255, G = R \pm (0, 1xR), B = R \pm (0, 1xR)$
<i>Land and empty land</i>	$NDVI = 0 - 1$
<i>Grass and shrub</i>	$NDVI = 0.15 - 0.3$
<i>Tropical forest</i>	$NDVI = 0.35 - 0.8$

Purpose of NDVI is classifying land use of every pixel data. The result will be matched with Horton coefficient as attribute of every pixel for running simulation of urban flood. From table 1, water and cloud have overlap range. Without RGB, water and cloud similar for range -0.3 to 1. To confirm the result, additional RGB method is needed. Then water and cloud could be classified as shown in figure 6a. The figure shows differences between cloud (white), water (blue), and land (red).

The next step is proceeding with DEM from satellite data. DEM is digital model or 3D representation of a terrain's surface. In this case, DEM represent height of every pixel by gray scale from 0-255, which 0 is the lowest area and 255 is highest area. For Surabaya, terrain height was in between 0-50 m above the sea. Figure 6 (b) shows the DEM of Surabaya. The difference of terrain is shown as differences of gray scale.

The next figure (figure 6(c)) shows drainage network maps. Drainage is the infrastructure that assists the water flow from the surface into receiving bodies of water, or artificial recharge facilities. Flood control is the facility to control the water level to avoid inundation. In this case, flood control is represented by water pump system namely Bozem. Receiving water body is a river, lake, or ocean that receives flow from the urban drainage system. Bozem and receiving water body are combined to perform map of city drainage map. We use the latest Surabaya map from Surabaya city mayor's office, dated on 2012. Classification and hierarchy of canal is denoted as follows: 0 is not a river, 1 is river, 2 is dot pumps, 3 is bozem, 4 is secondary canal, 5 is primary canal. This value is performed in matrix for simulating water volume in drainage system.

Last step is combining three maps in order to obtain appropriate drainage network coordinates, elevation map (DEM) and land classification map. Finally, the last physical data is the overlaid in the city of Surabaya map gathered from the Google map.

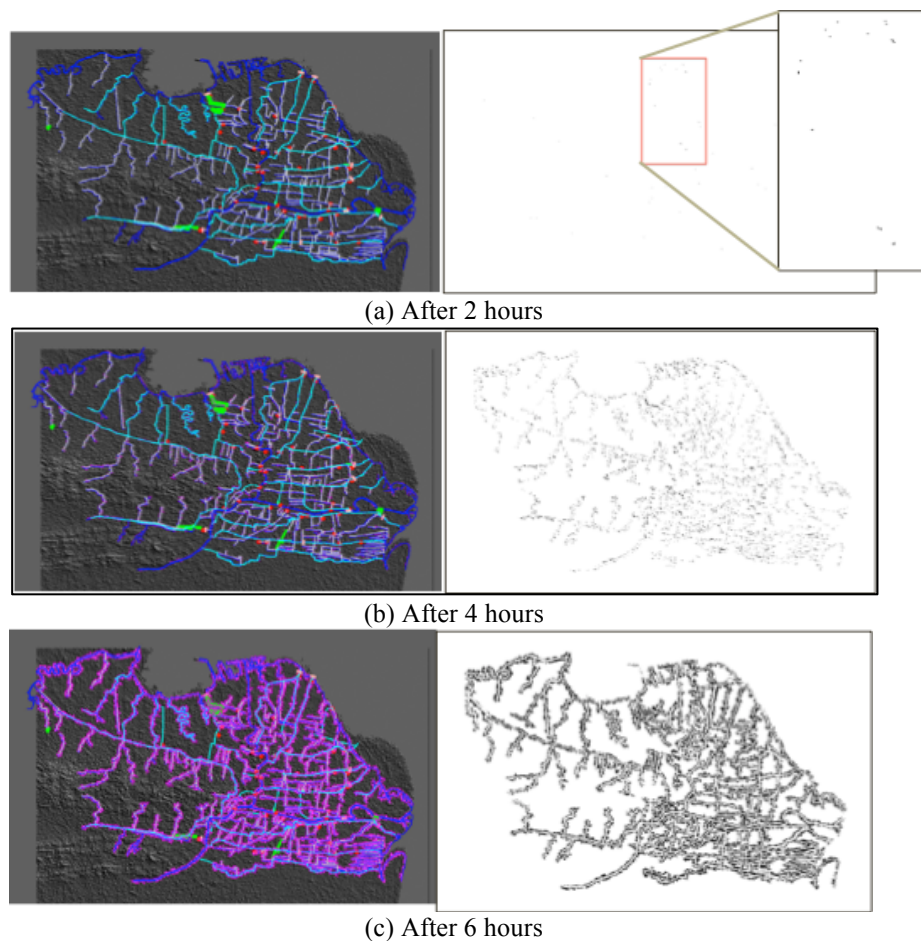


This map has been put over to satellite maps, topographic maps, and elevation, and drainage maps to get combination map. The result of the merger is shown in Figure 6d.

### 3.2. Flood Spreading

In this study, data is gathered from the real data satellite of Surabaya and data of one node sensor network rainfall and water level. The rainfall measure data and rain location is being input for flood mitigation systems. Then the simulation is conducted based on equation (1), (2) and (3) for every cell. These equations will calculate the  $R(t, x, y)$  on every cell. If  $R(t, x, y) > 0$  then rule of Cellular Automata (CA) is fulfilled. The CA is differential equation with differences of height as variable. In this study, the differences of height are obtained from differences of value in DEM from one cell to the neighbor in diamond pattern. Output of CA is direction and amount of water flow.

In this simulation, rainfall data is 250 mm as a representation of the most torrential rain in 2011, with length of rain time is 6 hours. The results are shown in figure 7.

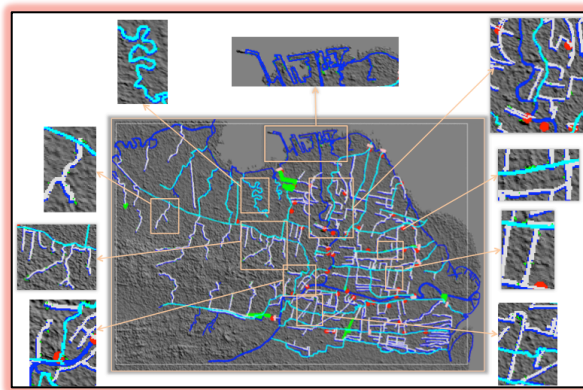


**Figure 7.** Simulation result for 2 hours-range

The left side of figure 7 shows simulation result of spreading after combined with map, while the right side shows simulation result before combined with map. The result shows that spreading area mostly starting from peripherals of drainage system to the certain area, in this case Central of Surabaya. This pattern also happened in several big cities with same characteristic like Bombay (India)[8], Jakarta (Indonesia), and Brisbane (Australia)[6]. This is the specific characteristic of pure urban flood that is rare happened in another type of flood such as flash flood, coastal flood or river flood. In the urban flood, changes in land use are in line with change in infiltration of soil. The lands change from soil to building or road, where the water-absorbing capacity is low [19]. Therefore in case of city with flat area, it is make a sense that the flood leads to spread in down town. This result is important for planning real time evacuation system during flood session.

Another interesting result from the simulation is fact that spreading is occurs nonlinear by time In first 2 hours, only several small inundation point, but 2 hours later it become larger more than twice and the in the third 2 hours almost all area is inundated. It means response of disaster in urban flood should be conduct as early as possible before flood happened. Otherwise, cost of the disaster will be increase dramatically [20].

In term of infrastructure, simulation result show association between urban flood and drainage system as illustrated in figure 8.

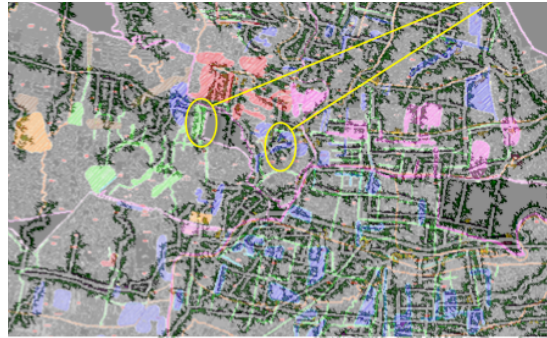


**Figure 8.** Simulation result by 2 hours

In figure 8, most of inundation point in Surabaya starting in area where secondary and primary canal meet. Theoretically, when water from larger surface section is entering the smaller one, flow of water will increase causing overflow in this area [21]. This result could be considered in planning of infrastructure in the future.

This simulation also useful for calculating the inundated area by counting the number of inundated pixel from DEM data. A pixel in DEM data represents 30 meters by 30 meters areas. The height of the water in each cell is calculated by subtracting DEM data of the highest cell inundated to DEM data of the evaluated cell.

As a validation, simulation result is compared with flood history map of Surabaya, 2011. The result is shown in figure 9.



**Figure 9.** Simulation result is compared with flood history map of Surabaya

This result is captured in downtown of Surabaya area. The background (white transparent) is history map while the dark one is simulation result. The result shows that pattern is similar especially in some areas near by drainage system but failed in another area. In this study, simulation is run with assumption rainfall event is happened uniformly for all area. But in fact, as shown in history map, some event of rainfall occur non uniform causing primary inundation. In the future, simulation in flood management should include data from many sensors in aggregate as well as in waste water management [22].

#### **4. Conclusion and Future Work**

Integration between physical sensor and modeling as abstract layer for urban flood mitigation has already developed. This simulation shows that CPH proposed system is very useful for real time response during disaster and planning in the future. In this paper, we proposed system includes physical layer that consist of sensors rainfall and river water levels and satellite sensors, abstract layer consist of flood modeling, interaction with human. We show that it become promising systems in the future for supporting urban flood management.

In critical situations, decisions must be fast and accurate especially in specific area such as downtown or Central of Business District; therefore this system should be improved by increasing the number of sensors and optimizing placement of sensors all over Surabaya. However, this system is not capable to manage very fast flooding processes like tsunami.

Future prospects of CPH for disaster is a dynamic system that is integrated, real time and, based on local knowledge, because it represent interaction between human and nature. Impact derived from the development of CPH-based disaster management system is a reliable system, privacy, easy to develop and support national self-reliance in disaster management. The strategy includes improvement in technical aspect and collaboration among researcher to share knowledge in disaster management.

#### *Acknowledgement*

This work is supported by the Ministry of Education and Culture of the Republic of Indonesia and the Electronics Engineering Polytechnic Institute of Surabaya.

## References

- [1] Data dan Informasi Bencana Indonesia, <http://dibi.bnpb.go.id>
- [2] Raghav Tripath, Sidharth Krishnan Sengupta, Adarsh Patra Heejun Chang, Il Won Jung, 2014, Climate change, urban development, and community perception of an extreme flood: A case study of Vernonia, Oregon, USA, *Applied Geography*, Volume 46, January 2014, Pages 137–146
- [3] V Notaro, M De Marchis, C M Fontanazza, G. La Loggia, V Puleo, G Freni, 2014, The Effect of Damage Functions on Urban Flood Damage Appraisal, *Procedia Engineering*, Volume 70, 2014, Pages 1251–1260
- [4] Harry Storch, Nigel K. Downes, 2011, A scenario-based approach to assess Ho Chi Minh City's urban development strategies against the impact of climate change, *Cities* Volume 28, Issue 6, December 2011, Pages 517–526
- [5] Takemoto, Shoko, 2011, Moving towards climate-smart flood management in Bangkok and Tokyo, MIT Thesis, <http://hdl.handle.net/1721.1/67243>
- [6] Espada Jr., Rodolfo and Apan, Armando and McDougall, Kevin, 2012, Spatial modelling of adaptation strategies for urban built infrastructures exposed to flood hazards., *Queensland Surveying and Spatial Conference: The Future of Surveying and Spatial Science is Open (QSSC 2012)*, 13-14 Sep 2012, Brisbane
- [7] Badan Perencanaan Pembangunan Nasional Indonesia 2014, Kebijakan Penanggulangan Banjir Indonesia, [http://www.bappenas.go.id/files/5913/4986/1931/20081123002641\\_\\_1.pdf](http://www.bappenas.go.id/files/5913/4986/1931/20081123002641__1.pdf), accessed on February 12th, 2014
- [8] Shahapure, S Eldho, Rao E, 2011, Flood Simulation in an Urban Catchment of Navi Mumbai City with Detention Pond and Tidal Effect Using FEM, GIS and remote Sensing, *J Waterway, Port, Coastal, Ocean Eng* 137(6) page 286-299
- [9] Schellart, Alma et al, 2011, Urban pluvial flood modelling with real time rainfall information – UK case studies, 12th International Conference on Urban Drainage, Porto Alegre, Brazil, 11-16 September 2011
- [10] Anne N. Glucker, Peter P.J. Driessen, Arend Kolhoff, Hens A.C. Runhaar, 2011, Public participation in environmental Impact Assessment : why, who and how, *Environmental Impact Assessment Review* Vol 43 November 2013 pages 104-111
- [11] Cahyono Susetyo, 2009, Urban Flood Management in Surabaya City : Anticipating Changes in Brantas River System, Master Thesis, [http://www.itc.nl/library/papers\\_2008/msc/upm/cahyono.pdf](http://www.itc.nl/library/papers_2008/msc/upm/cahyono.pdf)
- [12] Erol Gelenbe, Gökçe Gorbil, Fang-Jing Wu, Emergency Cyber-Physical-Human System, *International Conference on Computer Communications and Networks*, 2012.
- [13] A Ames et al, Human-Cyber-Physical Systems for Emergency Response, *International Conference on Intelligent Robot and Systems*, 2008.
- [14] Erol Gelenbe, Fang-Jing Wu, Large scale simulation for human evacuation and rescue, *Computers & Mathematics with Applications* Volume 64, Issue 12, Pages 3869–3880, 2012.
- [15] Nageswara S.V. Rao, Y. Narahari, C.E. Veni Madhavan, David K.Y. Yau, Chris Y.T. Ma, *Handbook on Securing Cyber-Physical Critical Infrastructure*, Chapter 3 – An Analytical Framework for Cyber-Physical Networks, Pages 55–72, 2012.
- [16] Jian Chen et al, A GIS-based model for urban flood inundation, *Journal of Hydrology*, Volume 373, Issues 1–2, Pages 184–192, 30 June 2009.

- [17] N.Y. Soma, J.P. Melo, 2006, On irreversibility of von Neumann additive cellular automata on grids, *Discrete Applied Mathematics*, Volume 154, Issue 5, 1 April 2006, Pages 861-866
- [18] R. K. Gupta, D. Vijayan, T.S. Prasad, N.Ch. Tirumaladevi, 2000, Role of bandwidth in computation of NDVI from landsat TM and NOAA AVHRR bands, *Advances in Space Research*, Volume 26, Issue 7, 2000, Pages 1141-1144
- [19] Howard Wheatler, Edward Evans, 2009, Land use, water management and future flood risk, *Land Use Policy*, Volume 26, Supplement 1, December 2009, Pages S251-S264
- [20] Q Zhou, P.S. Mikkelsen, K. Halsnæs, K. Arnbjerg-Nielsen, 2012, Framework for economic pluvial flood risk assessment considering climate change effects and adaptation benefits, *Journal of Hydrology*, Volumes 414–415, 11 January 2012, Pages 539-549
- [21] E. Mignot, A. Paquier, S. Haider, 2006, Modeling floods in a dense urban area using 2D shallow water equations, *Journal of Hydrology*, Volume 327, Issues 1–2, 30 July 2006, Pages 186-199
- [22] Víctor-M. Sempere-Payá, Salvador Santonja-Climent, 2012, Integrated sensor and management system for waste water network and prevention of critical situations, *Computers, Environment and Urban Systems*, Volume 36, Issue 1, January 2012, Pages 65-80

# Design and Prototypical Implementation of an Integrated Graph-Based Conceptual Data Model

Matthias Sedlmeier, Martin Gogolla

**Abstract.** The paper introduces a new, comprehensive integrated conceptual data model in a precise way. Central language features cover core modeling concepts as classification, membership, inheritance, interfaces, structured types, aliasing, aggregation, composition, constraints, clustering and relationships. Schema states are thought of as being realized as graphs with appropriate navigation options. A prototypical graph database implementation and accompanying examples are discussed.

**Keywords.** conceptual data model, graph based storage, graph database, conceptual modeling

## 1. Introduction

Information technology depicts a fundamental component of modern organizations in areas like administration, service and production. Well working electronic data processing is able to raise the efficiency of administration and business processes [11] by supporting employees accomplishing their tasks. Poorly designed information systems however may disturb work flows and lead to unmotivated labor [23]. The design of a good information system is challenging, because its quality depends heavily on the sound analysis of the requirements [33]. These are normally specified in cooperation with the representatives of all later end user groups to get a comprehensive impression of the needs.

An information system should be able to cover the information needs of a particular organization. Therefore, it must have the ability to store structured data, to calculate and derive new data and output data in a human readable way. One can assume, that the need of certain information is always linked to a specific task [18]. As information systems support specific tasks varying from organization to organization, their requirements also differ. The exact requirements are usually determined in the form of a data model holding a precise representation of the domain.

The ability of human intellect to abstract from reality establishes the fundament for designing data models. The process of abstraction implies the concentration on designated characteristics of a natural object and at the same time the abandonment of detailed examination of all recognizable attributes. Abstraction is hence an act of simplification or generalization. The design of a good data model demands an accurate analysis of the application domain in such an extent that the correct subset of all observed aspects is captured and represented appropriately.

To represent real domain aspects, one may use graphical language elements, which stand for objects, attributes and relationships and therefore receive a semantic denotation. Specific rules of combining and linking those graphical artifacts are defined by a corresponding syntax. In addition to the representation of natural considered objects, data models also describe ideal constructs such as *divorce*. Such standardization of a modeling language gives designers the ability to create models, which can be understood by anyone knowing how to interpret the language. The definition of a syntax also enables designers to apply formal model modifications as well as model transformations automatically. The possibility to create new models by transformation respectively translation is crucial for the design process of an information system, where each development stage claims models of different abstraction levels.

The controlled storage of data is usually done with the aid of a special software often denoted by *database system*. A database system consists of one component responsible for the physical data storage and further components dealing with data processing and access control [8]. The first component is also known as *database* and the second component is denoted as *database management system*. The design of a database as part of an information systems is divided into multiple phases, whose results end in corresponding documents. Depending on the current design stage, the domain artifacts are represented in different models, which vary in their way of expressing domain issues like structured types or relationships.

In the course of creating the requirements definition, one initially elaborates a conceptual model, which is also called a conceptual database schema. This model is typically worked out as an ER [7] or UML [31] diagram and serves as background for a logical database schema, which regards the concrete operating mode of the database model. The last step contains the translation of the logical database schema into a physical model.

The most frequently used database model is the relational one, which uses tables to represent the elements of the conceptual schema. But modeling domain aspects via tables often proves itself as complicated and insufficient, because tables do not allow the *direct* representation of domain facets. A solution to face these problems presents the design of an integrated semantic data model as a synthesis between conceptual, logical and physical model. For this approach, parts of the language features of the ER respectively EER [30] model and the UML class model are used. Moreover, a main memory resistant graph-based storage approach is chosen to make data records persistent. Abiteboul and Hull follow a similar idea in their work about the design of a semantic database model [1]. The introduced integrated semantic data model is not just another conceptual data modeling language. It is also considered as a type graph, from which database states can be directly derived by establishing instance graphs. A similar approach is used in [2], while [21] discusses search algorithms for conceptual graph databases.

These can also be found in medical computing [10], data warehousing [19], bioinformatics [12] and geographic information systems [24]. A comparison between graph databases is presented in [16] and [20], which lays the focus on performance. In [3] another comparison of current graph databases is provided. Graph databases are also subject of researches related to big data [25] and data mining [28].

The rest of this paper is structured in 5 sections. In section 2, some challenges are discussed. Section 3 introduces the designed integrated conceptual language called TEGeL. Section number 4 deals with the implemented database prototype called Lhasa

DB. And in section 5, a short evaluation is given. The paper closes with a conclusion and some ideas for future work in section 6.

## **2. Challenges**

### *2.1. Requirements Analysis*

The requirements definition is an important document generated in an early phase of the software development process. If an information system is designed, a lot of requirements are related to the question, which data must be stored in which way. Therefore, the first step contains a sound analysis of the domain. This task can be conducted by using graphical models prior to textual descriptions, which usually express the requirements in a more clearly and easy way. To cover every important aspect, everyone using the system later on, has to proclaim his information needs. In most cases these potential users are neither software developers nor do they have a technical background. Upon this, one must emanate from the fact, that all kinds of different careers have its own special requirements, which must be formulated in one comprehensible model. Therefore the chosen language must ensure, that everyone is able to understand its syntax and semantic. If this goal is reached, the conceptual model can be applied as a communication basis between developers and users [6].

As the practice shows, only 30 percent of all users are aware of their needs. Over 40 percent of their requirements are indeed known and action determining, but not directly accessible and though cannot be formulated. Another 30 percent of all requirements are subconscious and must be triggered from the outside [27]. This fact must be kept in mind when choosing a good conceptual language, which must be understood easily while leaving room for frequent changes. Requirements can also vary systematically due to the selected development process. If an agile approach is given, frequent increments of the requirements definition are part of the development principle. Therefore, a good conceptual model has to support agile design cycles.

As we can see, requirement changes are not avoidable. In practice, they are the order of the day and ultimately part of agile development principles. Under these circumstances it is necessary to ensure, that alteration costs are minimized by including all changes as early and straightforward as possible. Alteration costs can increase quickly: once the system is under design, the costs of changes raise by a factor of 3 and during its development by a factor of 7. If there are requirements changes while testing, the alteration costs increase by a factor of 50 and after delivery by a factor of 100 [26].

### *2.2. Problems of Relational Modeling*

The data management component of an information system usually consists of a relational database, which is designed in three phases. The first phase includes the abstraction of the application domain in form of a conceptual model. This conceptual model is then transformed to a logical model, from which the physical model is derived. The conceptual model is often implemented as ER respectively UML diagram, which is able to represent facts like relationships directly. When it comes to relational databases, the conceptual model is used to derive a relational schema using tables with appropriate



columns. Further steps possibly contain normalization and synthesis. In the last phase this relational abstraction is taken and translated to corresponding SQL statements.

At the end of the pictured process, there are at least three models in different languages describing a single application domain. If any of these models is changed, all other models must be updated to prevent inconsistent representations. On the one hand, the correctness of the conceptual model is important, because it is the communication basis between all stakeholders. On the other hand, the derived models should always be synchronized with the conceptual model to represent the application domain accurately. It must be pointed out here, that it is always technically possible to manipulate every model in itself without having adopted those changes automatically to the remaining models. One explicitly must take care about the semantic cohesion between all models after every change. This synchronization contains continuous model transformation in different languages and is possibly lossy, if there is no direct mapping between the used language features. For this reason, a transformation may also not be reversible impeding the whole synchronization process.

As said before, the conceptual model plays an important role with regard to stakeholder communication. That is why it is created as an ER or UML diagram, which is much more comprehensible for technical lays than relational schemata. This is connected with the visual representation of domain facets and the kind of expressing relationships. In ER or UML models, these are illustrated by edges, which can be seen as the most direct semantic mapping possible.

The representation of relationships by tables falls far short of this simplicity, because one must use foreign keys and join tables to express the connection of two or more objects. This indirect description does not reflect the fact, that the connection of domain objects rather correspond to graph structures as it is clearly apparent in the conceptual model. This representation problem arises from the fact, that tables are flat and thus not able to express links and structures in a native way. Further problems occurring from this deficit of straightforward mapping are discussed now.

Domain objects usually have attributes for detailed specification. If those attributes are flat, they can be represented through suitable named table columns. Structured attributes however shape hierarchies, which can only be modeled indirectly using the relational join. This way must also be gone, if there are multiple attributes with an arbitrary maximum cardinality. Another problem depicts the Object-Relational Impedance Mismatch [5], which occurs using object-oriented business logic. Data structures, which were flattened for relational storage, now must be recovered to their structured versions. After processing, the data records are flattened again and so forth. This continuous translation induces additional computing costs. The mapping of relational data to object structures is often realized by a object-relational mapper, which can be added to an information system as further layer. Here, one must bear in mind, that using an object-relational mapper introduces an additional need for synchronization, because the business logic always works on a copy of the original data. Another problem is related to the concept of inheritance, which can be simulated with patterns like Class Table Inheritance, Concrete Table Inheritance or Single Table Inheritance [9], but which is not a native component and therefore offers no direct representation.

### 3. Introducing TEGeL

The controlled storage of data requires a formal model revealing which data must be saved, which connections exist and which constraints must be considered when database states are established. The following sections describe such a model as a *type graph*.

The central element of a type graph depicts the *entity type* as an abstraction of an object or idea. An entity type allows the instantiation of *entities*, which reside in a potential database state with a proper existence. Entity types can be categorized by *entity type classes*, which is expressed by a *classification relation*. An instantiation of entity type classes is not possible, hence they are marked as *virtual*.

Entity types are described in detail by *annotation types*, which represent attributes. Entity types and annotation types are connected by *membership relations*. Annotation types itself can equally be connected by membership relations, if structured attributes are modeled. Instantiated entity types, thus entities, hold accordingly instantiated annotation types called *annotations*. There are different *modes* of annotation types, which will be discussed later.

Besides the *reuse* of annotation types by multiple entity types, the inheritance of attributes is provided by introducing the concept of a *super entity type*. It equals an entity type except for the fact, that an instantiation is not possible. Super entity types may inherit from multiple other super entity types, which is realized by an *inheritance relation*. In addition to inheritance, one can use the concept of *entity type interfaces*, which also delivers extra attributes for entity respectively super entity types. The connection is established by an *implementation relation*. Entity type interfaces are not able to inherit from super entity types or to implement other entity type interfaces. The *clustering relation* enables the designer furthermore to categorize entity types, entity type classes, super entity types and entity type interfaces in named *clusters*.

The type graph does not only contains type information, but also rules for relations and constraints. The integrity rules are generally denoted as *integrity constraints* and decompose in *value constraints* for data type restrictions and *structural constraints* for structural requirements. Modeling relationships is also possible by using *relationship types* in combination with a special annotation type mode called *aggregation*. In a database state, relationship types become *relationships* interconnected with *aggregation annotations*.

The introduced type graph represents an integrated conceptual language, which will be called Tibet Entity Graph Language (TEGeL) from now on. Specific nodes and edges corresponding to the mentioned concepts above will be described and illustrated<sup>1</sup> in the following sections.

#### 3.1. Classification and Membership

Figure 1 introduces the classification as well as the *default* membership concept for (reused) *annotation type nodes* and additionally shows comparable ER and UML models. The given schema models a kind of water sports world with two domain concepts, which are represented as entity type nodes: notably Lido and SportsBath. These *entity type nodes* are grouped together by an *entity type class node* named SportsFacility and share two annotation type nodes Name and Location, which represent *plain* attributes.

<sup>1</sup>All example schemata were created with the yEd Graph Editor.

These annotation type nodes are subject to integrity constraints as their possible instance values may only be elements of the string data type. The connection between entity type and annotation type nodes is generally established through a *membership edge*, which has different modes. In this case, the membership is qualified as *required* and *default*. Other modes are discussed later.

Sharing or reusing annotation type somehow breaks with the concept of encapsulation, but it provides an advantage towards integrity, because one can now assume, that an attribute with equal naming (irrespective of the local context) describes the same application domain artifact. The redundant definition of annotation types and their integrity constraints may thus be prevented.

Both entity types have distinct attributes named `NoSlides`, for the number of slides, `ChildrensPool`, which is set to true, if there is a pool for children and `StopWatch`, which is set to true, if there is a stopwatch to take the lap times. These attributes are also restricted when it comes to value assignment. There are two orders of value constraints: the data type is a *first order value restriction*, but it is possible to define *second order value restrictions* in form of simple conditions or regular expressions as it is apparent for the annotation type nodes `NoSlides` respectively `Name`. TEGeL is planned to support a subset of the ordinary atomic XML Schema [22] data types including *string*, *decimal*, *integer*, *float*, *boolean*, *date* and *time*. Sequences, sets and structured types can be directly modeled via TEGeL.

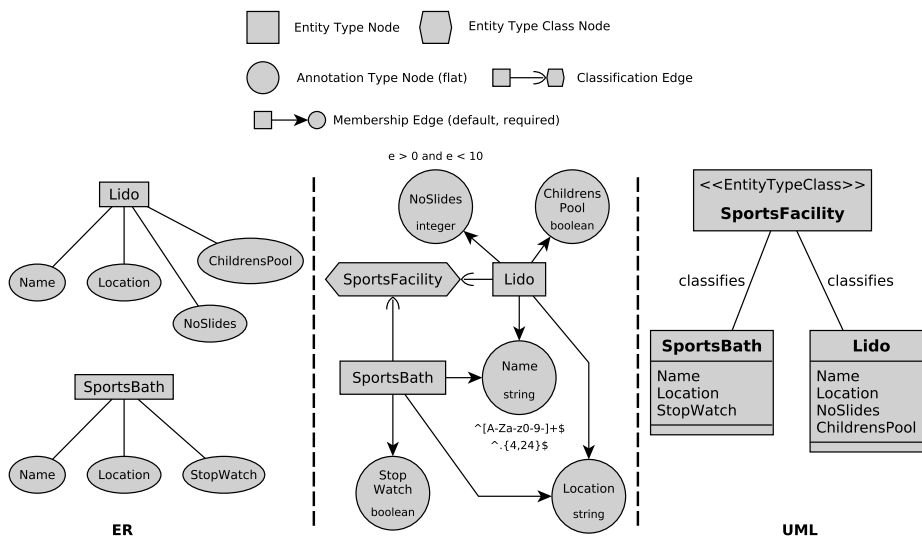


Figure 1. Classification and Membership

Entity type nodes are drawn as rectangles with sharp edges, whereas entity type class nodes are represented by hexagons. The displayed annotation type nodes are in plain mode, so they are drawn as circles. Each node must at least have a unique identifier and may carry additional information, called *decorations*. In the given example, the indication of first and second order value restrictions can be considered as decorations as well as the data type indication. Custom decorations may also be sensible, but will not

be discussed here. As one can see later, also edges hold decorations, which are called *modifiers*.

The membership relation between both entity types and their annotation type nodes can be directly drawn as edges with normal arrow heads directed from the entity types towards their attributes. The classification edges towards the entity type class node `SportsFacility` are also solid, but show open concave arrow heads.

### 3.2. Inheritance, Interfaces and Structured Types

Figure 2 again models a kind of sports world and introduces the concept of inheritance. The already known entity type nodes `SportsBath` and `Lido` inherit all annotation type nodes of the super entity type `RecreationFacility`, namely `Name`, `Location` and `Admission`. The first two attributes are already familiar from the above example schema, where they were modeled as *reused* annotation type nodes. A *super entity type node* is drawn as a trapezoid connecting entity type nodes by *inheritance edges* with a white triangle arrow head. The third annotation type node is a structured attribute called `Admission` holding two plain annotation type nodes `AdmissionType` and `Amount`. Annotation types in *structured mode* are drawn in rectangles with round corners and have no proper data types as they themselves build new ones.

Next to the `Admission` annotation type node, there is another structured attribute, which forms a new data type representing a `Pool`. As we can see, *nested* structured types can be composed of other structured types forming a hierarchy. Regarding the membership between `SportsBath` and `Dimension`, one can also see, that structured annotation type nodes can be reused, even if they are part of a substructure. Furthermore, we see, that membership edges can be *optional* if they are drawn in a dashed way. If annotation types have an optional membership, they are allowed to be absent when instances are built.

Another point is the indication of a *membership cardinality*, like it is shown for the attribute `Admission`. Membership cardinalities provide a mechanism to control the quantity of established attribute instances and induce a *multi membership*. Together with the indication of optionality, it represents another way of defining integrity constraints.

Additionally, functional dependencies can be established by declaring memberships as *unique* as it is done with the plain annotation type node `GeoCoordinates`. If a multiple attribute membership edge (one with cardinality indication) is modified as unique, the whole collection of attributes instances has to be unique concerning the element values as well as their order (if it is not a set). Multiple attributes can also be attached as *distinct*, which means, that every element within the collection must be unique.

A further important concept depicts the entity type interface. It is applied to be sure, that entity types hold certain attributes, which are necessary for the processing of derived instances. Expressing interfaces by modeling corresponding entity super types is, although technically possible, semantically misleading, because a super entity type usually represents the same domain object on a higher abstraction layer. This is also the reason, why entity types interfaces neither implement other entity type interfaces nor inherit from any super entity types. Conversely, super entity types possibly fulfill interfaces as it is shown on the example schema, where `RecreationFacility` is connected to entity type interface `Visitable` by an *implementation edge* with a dot arrow head. The *entity type interface node* itself is drawn as an upside down trapezoid.

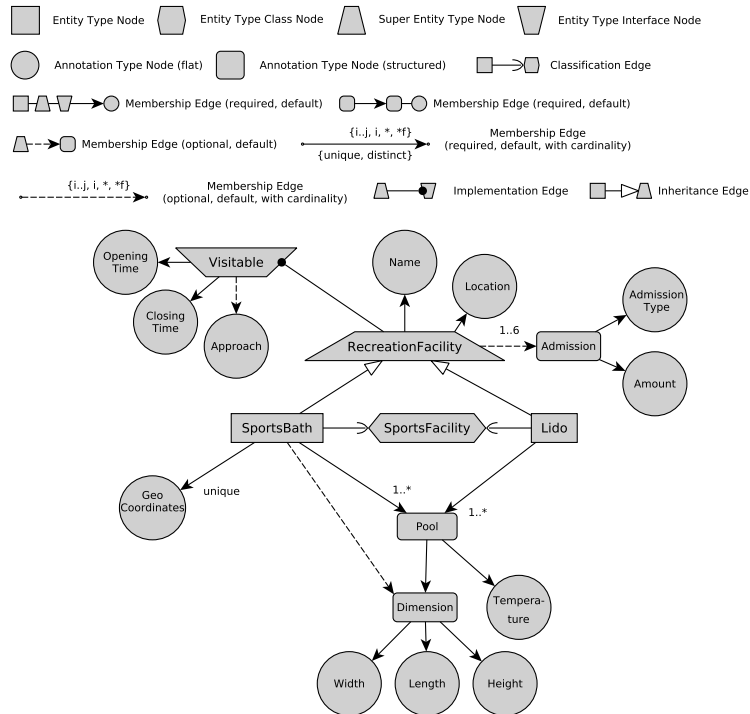


Figure 2. Inheritance, Interfaces and Structured Types

### 3.3. Aliasing, Constraint Groups, Aggregation and Clustering

Figure 3 introduces *alias annotation type nodes*, which are drawn as diamonds. By means of those attributes, one can use existing annotation type nodes while renaming them. The example shows a possible application for this linguistic feature. Both alias annotation type nodes `BeginDate` and `EndDate` only have a slightly different semantic notion. Here, this feature realizes a fine specialization upon the `Date` attribute, whose values additionally must comply with the condition  $e > 01/01/1980$ , where  $e$  is a possible concrete date.

So far, one can only make statements about the required or optional existence of attributes. The introduction of *constraint group annotation types* allows the definition of more precise conditions. Assume that information about sporting events has to be stored. The example schema shows a possible variant by modeling an entity type node called `SportingEvent` with a plain attribute `Title`. There is also a member called `Venue` representing a constraint group annotation type node, which is connected by a *restriction membership edge* with an opened convex arrow head. This constellation demands, that, when a database state is constituted, there can either be a connection to the aggregation annotation node (which will be discussed later) `SportsHall` or to `Stadium` as the inbound edge modifier `XOR` indicates. The constraint group annotation type node itself is drawn as a parallelogram. Besides the logical operators `OR`, `AND` and `XOR` also relational operators like `>`, `<`, `=`, `>=`, `<=` and `!=` can be used, if the member data types provide support.

Till now, the presented concepts do not allow to correlate entity types. This serious restraint is cured with the introduction of *aggregation annotation type nodes*. In the field of data modeling, relationships are often denoted as associations and usually allow the connection between one, two or more objects. Thereby, the exact semantic of those associations is not fixed. In practice, one must frequently express, that objects have a whole-part connection to model hierarchical structures or to formulate existence conditions. For the first case, the *aggregation* can be applied representing a specialization of an association. This relation is usually directed, so that there exists a whole, which is here called *aggregate* and subordinated parts named *components*. There is no existential binding between the aggregate and its components. If the aggregate's life cycle terminates, the components are released from the relation and persist.

For the second case, the concept of *composition* comes into question, which enables the designer to connect entity types existentially. Compared to aggregation, the existence of all components are annihilated, if the aggregate is deleted. The composition can be also seen as a specialization of the association. Following the naming conventions of UML, the aggregation mode is either *shared* for the representation of aggregation or *composite* for compositions.

In the previous example schema, the pool artifact was designed as a structured annotation type node, which made it an integral part of a lido respectively sports bath. If we want to model the pool as an independent concept, the composition comes into play and the pool is then modeled as an entity type node. We express the fact, that a pool should be bound to the existence of its lido or sports bath and won't be shared between multiple aggregates. The entity type `Pool` is connected to `Lido` by an aggregation annotation type node, which itself is linked to the `Lido` entity type node through a specialized *composition membership edge*, which shows a black diamond as arrow head. The edge direction is knowingly reversed in contrast to the UML notation. The cardinality indication shows, how many pool instances may be composed. The `SportsBath` entity type is also connected to the pool artifact by another aggregation node.

Additionally, one can determine a shared aggregation between `Lido` and `RubberMonster`, which is also established by an aggregation node through a *shared membership edge* drawn with a white diamond arrow head. This connection depicts, that a lido may offer a rubber monster, which is not existentially bounded. All aggregation annotation type nodes are linked to corresponding entity type nodes by aggregation edges equipped with crow arrow heads.

*Monomorphic aggregation* respectively composition is very inflexible, because it only allows to connect instances holding exactly the specified type. For this reason, TEGeL permits directing aggregation edges not only to entity types, but also to all other concepts except for annotation types. This feature is here called *polymorphic aggregation*. Aggregation annotation types also have identifiers, which can be seen as *target role names* for the attached entity types offering an additional navigation possibility.

Another language feature are *cluster nodes*, which enable designers to categorize entity types, super entity types, entity type interfaces as well as other clusters. A very limited example is given in the example schema. The `Attraction` cluster node connects the entity type node `SportingEvent` as well as `InflatableMonster` and the entity type interface node `Visitable`, which means, that every other type implementing this interface is also part of the cluster. In this case, `Lido` as well as `SportsBath` are transitive elements of `Attraction` via the `RecreationFacility` super type node.

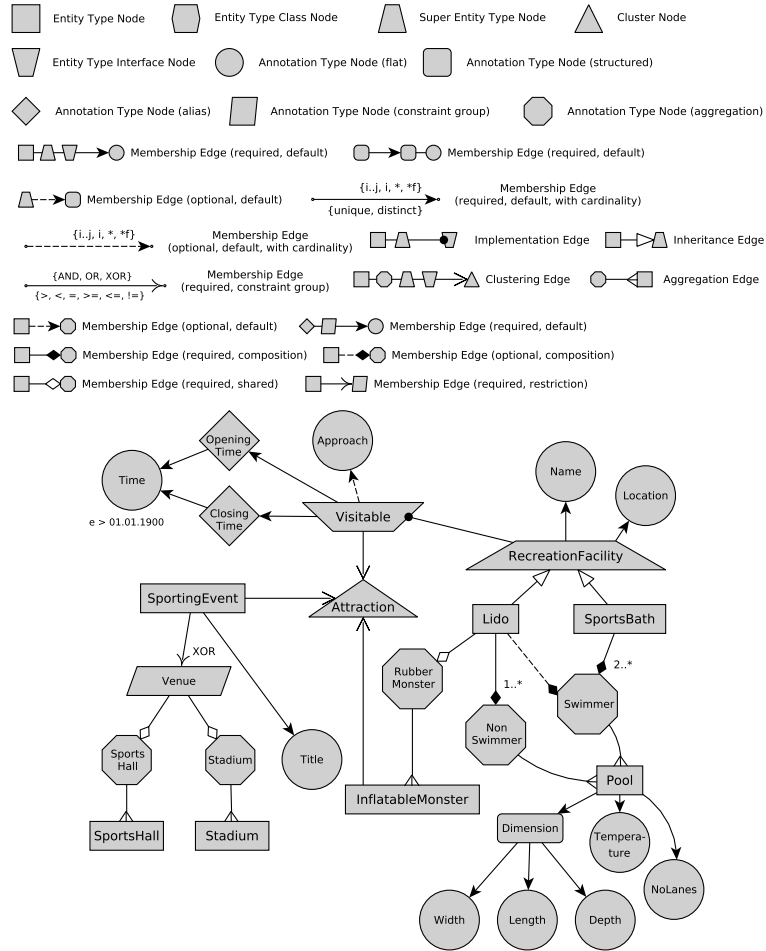


Figure 3. Aliasing, Constraint Groups, Aggregation and Clustering

### 3.4. Relationships and Hyper Relationships

Relationship types enable the designer to establish associations between entity types respectively relationship types. Relationship types are handled as specialization of entity types. In the context of a schema definition, one can specify which entity types respectively relationships types are connected and which quantitative participation constraints they must comply. Therefore the membership cardinality as well as the *participation cardinality* is used.

If cardinalities are indicated, one must differ, if relationships or entities are counted. The participation cardinality restricts, how often an entity takes part in a specific relationship. Conversely, the membership cardinality counts how many entity instances participate in an aggregation. So, one can basically count (a) the number of participations of entities in relationships (participation cardinality) or (b) the entities participating in an aggregation (membership cardinality).

Besides the indication of cardinalities, relationships have an arity defining, how many entity or relationship types are participating at all. Usually, relationships are binary, but they can basically have any arity greater than zero. Additionally, relationships are distinguished between their orders. First order relationships only connect entity types, while higher order relationships also connect first order relationships. In this paper, these higher order relationships described in [4] are named *hyper relationships*.

Usually, three kinds of binary relationships are distinguished. Consider  $A$  and  $B$  as entity sets. The 1:1 relationship describes a *one-to-one* mapping between an element of  $A$  and an element of  $B$ . The 1: $n$  relationship specifies a *one-to-many* mapping, where one element of  $A$  has a connection to  $n$  elements of  $B$ , but an element from  $B$  relates only to one element of  $A$ . Finally, there is the  $n$ : $m$  relationship, where  $n$  elements of  $A$  can be connected with  $m$  elements of  $B$  and vice versa.

Figure 4 shows three binary first order relationship nodes called *Membership*, *Offer* and *Execution* as well as one binary hyper relationship node named *Training* and finally one ternary first order relationship node denoted as *Contact*.

The relationship type *Membership* models the fact, that an athlete is a member in a gym. Given the membership and participation cardinalities, we can make the following statements: Firstly, one *Gym* instance has between 100 and 1000 members. Secondly, one athlete has any number of gyms, at which he is enrolled. Thirdly, one *Membership* instance holds an arbitrary number of gyms and fourthly, one *Membership* instance holds an arbitrary number of athletes in *fixed mode*. The fixed mode means, that for every new *Athlete* instance a new *Membership* instance is created to preserve ambiguity. Therefore, the left side membership cardinality of *Athlete* is internally always one and its participation cardinality is arbitrary. The right side membership cardinality of *Gym* is arbitrary, while its participation cardinality lies between 100 and 1000. This kind of representation is called the *compact representation of relationships*, as it doesn't establish a relationship instance for every athlete-gym tuple.

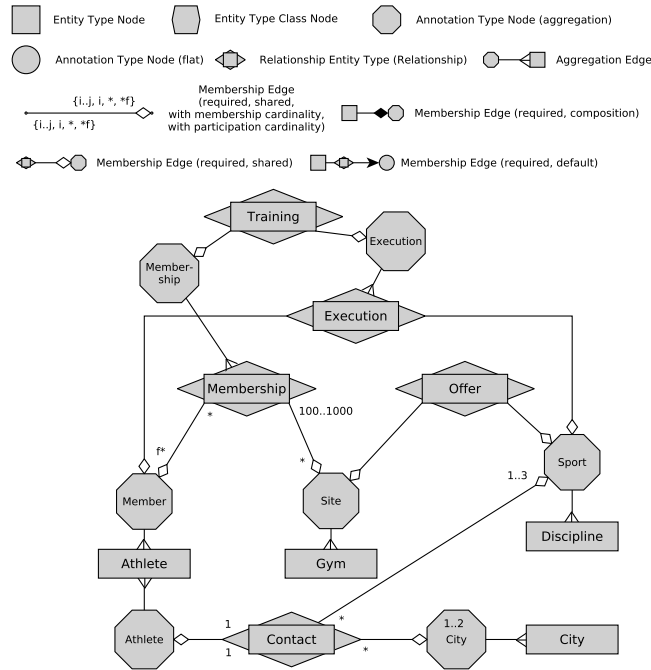
Another connection is established by the *Contact* relationship type. This relation models the fact, that an athlete acts as a contact person for one to three sports in one to two cities. An athlete can act as contact person only once, whereas sports and cities can be included any times. A relationship node is drawn as a combination of an oblong diamond overlaid by a rectangle.

#### 4. Prototypical Implementation

This section deals with the design of the implemented prototype database, which is called *Lhasa DB*. The main aspects of this concept are represented by the *indoor* and *outdoor* metaphors.

As *Lhasa DB* persecutes a main memory approach, it is important to realize, that entities always reside in the main memory, no matter if they are actually *stored*. That is the reason for creating a virtual distinction between entities, which are indeed in the main memory, but not officially stored (thus *transient*) and those entities, which are marked as stored and thus officially *persistent* (and still in the main memory). In the given context, persistent means, that entities have passed through the regular *storing process*, which brings them from outdoor to indoor. This event is called *entity transition* or *entity crossing*.





**Figure 4.** Relationships and Hyper Relationships

Entities, which do not carry the status stored are in a virtual region called *outdoor* or *outdoor sphere*. Entities carrying the stored status are in a virtual region called *indoor* or *indoor sphere*. The entity transition is triggered by a storing process letting an outdoor entity cross the *sphere frontier*. Entities before this process are transient, entities after this operation are considered persistent. Therefore, the common definition of *persistent* is changed not demanding a permanent storage outside of the main memory any more.

The latter aspect is realized by the *materialization process*, which backs up the persistent database state in the main memory to a permanent storage device. This procedure can be synchronized with the main memory storing process by enabling the *write-through* mode.

The crossing of an entity depends on whether it complies to the given schema. Therefore, the storing process initially checks, whether all constraints are fulfilled and the entity is composed correctly, that means, if a correct graph structure with all necessary nodes and edges is given. If that is the case, the entity and its annotations transit to the indoor sphere, which holds the current database state. This database state is represented through the *instance space* consisting of *instance graphs*, which are derived from the type graph and are accessible through entity and annotation indices. The crossing process also contains the *unification operation* stating whether there are value occurrences, which were stored before. If that is the case, these values are referenced instead to save main memory space and to fasten look-ups.

Instance nodes usually are connected in a *bidirectional* manner, that is a node  $n_1$  has a directed outgoing reference to  $n_2$  and  $n_2$  an ingoing reference from node  $n_1$ . So usually, one will spot two edges realized by trivial memory references. But that is not the

case when it comes to unification or *shallow entities*, which are discussed later. Here, a *unidirectional* connection is used.

Assume, one wants to store a pool entity, whose type information is given in schema 3. In the first step, one must initially establish a living entity instance in the outdoor sphere. Therefore, the *prime* operation is used, which takes the desired entity type identifier as a parameter. After this is done, attribute values can be set accordingly to the schema information by instantiating corresponding annotations. Lhasa DB provides *meso operations* named *set*, *use* and *unset* to handle attribute values. The prime operations is one of the five Lhasa DB *PiCRUD macro operations* *prime*, *create*, *retrieve*, *update*, *destroy*, which will be discussed in detail later.

If an attribute value is assigned, usually an annotation node is build and provided with its value. In this case, the entity and the annotation node have a bidirectional connection. Both nodes are still in the outdoor sphere and thus not part of a potential indoor database state. When it comes to unification, the described process shapes differently. Here, the set operation checks, whether there is already an annotation with the same type and value, which was stored before and thus can be detected indoor in the instance space. Under these circumstances, the indoor annotation is referenced by a unidirectional connection coming from the outdoor entity. This reference must be unidirectional and it must be directed from the outdoor sphere into the indoor sphere to not alter the current database state. This kind of unification is called *instant unification*. So, outdoor structures are fully connected, if the involved nodes are transient (not stored), else unidirectional connections are used. Persistent (stored) indoor structures are always fully connected.

After all attributes are set, the established entity itself has to be integrated into the database state. Therefore, one uses the create operation, which triggers the storing process. This process initially performs integrity checks, unifies those attributes, which were not unified before (*deferred unification*) and expresses the state integration as a series of Lhasa DB *micro operations*, which realize instance space index manipulations as well as edge reallocations. The integration of the outdoor graph structure into the database state is here described as the *weaving process*. So the create operation triggers the storing process, which overall includes (a) a validation process, (b) a final unification process, (c) a weaving process and at least (d) a materialization process, which was mentioned above.

So far, the first two macro operations are covered. The next operation to be mentioned is called retrieve. This operation reads a stored entity and makes it available to the outdoor sphere, where it can be manipulated by the database user applying the mentioned meso operations. These actions then may end up in calling the PiCRUD update operation, which again triggers the store operation.

If an entity is brought from indoor to outdoor again, there are two main challenges, which must be addressed: the integrity of the database state and the synchronization of parallel access, when the entity is updated. That is the reason for introducing the *shallow entity*, which only exists outdoor and holds a pointer to its indoor counterpart. If an entity is to be retrieved, initially a shallow copy is created. Secondly, all annotations of the original entity are also linked to the shallow copy in a unidirectional manner. Finally, this construct is returned to the outdoor sphere and then accessible by the user. Changes, which are made on the shallow copy, do not immediately change the indoor database state and are deferred until the update operation again initiates the storing process, which is synchronized by the Lhasa DB *Transition Dispatcher* and *Concurrency Controller*. If a shallow entity is stored, it is treated as a new entity except for the fact, that all its new

annotation nodes are reallocated to the existing original entity, whose previous annotation nodes are then detached. The shallow entity is then expunged. The last operation *destroy* deletes the desired entity by removing it from the instance space index. Its annotation nodes reside, if they are used by other instances.

Figure 5 shows the unification before and after the storing process. Assume  $x_2$  as an outdoor entity instance of type  $X$  and  $a_2$  as an outdoor annotation type node of type  $A$ . Node  $x_2$  and  $a_2$  are connected in a bidirectional way, as both nodes are outdoor and full graph navigation possibilities are given. Entity  $x_2$  has another attribute node  $b$  of type  $B$ , which resides indoor. This annotation types was attached in a unidirectional way during the instant unification process. One can navigate from the outdoor entity to this indoor attribute, but one cannot navigate from  $b$  to  $x_2$  as long as it is not stored. Creating a bidirectional connection between indoor and outdoor nodes is illegal, because alteration of  $b$  would change the database state. As the figure illustrates,  $b$  was originally brought to the indoor sphere by the stored entity  $x_1$ , which fully references another attribute node  $a_1$ . After the storing process, the instance space experienced an alteration and  $x_2$  is now part of the database state. It is now save to fully attach both nodes  $x_2$  and  $b$ .

As it is important to check, whether operations change the internal state, all Lhasa DB operations are classified as either *state invasive* or *state retaining*. The meso operations set, use and unset are state retaining because they never change an indoor attribute directly. For instance, a set operation for assigning a new value to a stored unidirectional linked annotation node always causes the instantiation of a new attribute instance with the same type. The set operation only changes the attribute value directly, if it resides in the outdoor sphere. The macro operations PiCRUD are partly state invasive and partly state retaining, the same applies to the micro operations.

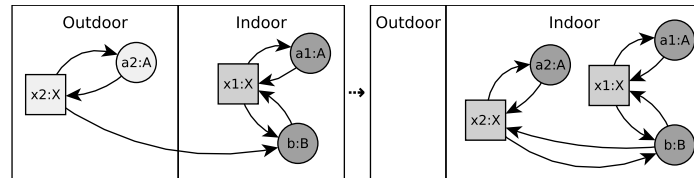


Figure 5. Unification before and after the Storing Process

## 5. BibTeX Experiment

### 5.1. The Scenario

The following experiment gives an impression about time and space requirement for establishing a database state. Based on a BibTeX example, a TEGeL schema is build and instantiated with given example data. The example schema is given in figure 6 and the example records consists of 205 Pub, 491 Keyword and 152 Author entities. Furthermore, it involves 1519 PubKeyword as well as 594 PubAuthor relationship nodes.

Based on these numbers and the given schema, one can calculate the hypothetical quantity of all instance nodes by counting all entity, attribute and relationship nodes required to establish a valid state. This calculation does neither consider the compact representation of relationships nor the unification.

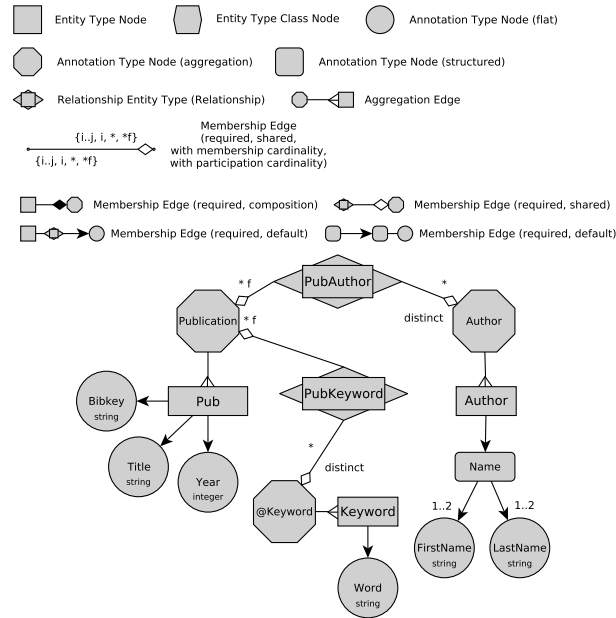


Figure 6. TEGeL BibTeX Example

So, one would completely establish 820 nodes for all Pub, 982 nodes for all Keyword and 925 nodes for all Author entities. Additionally, one must count 4557 nodes for all PubKeyword as well as 1782 nodes for all PubAuthor instances. By determining these figures, one must keep in mind, that multiple attributes always require an additional multi node and that relationships are basically modeled by additional aggregation nodes, which increase the final number of nodes seriously. After all, this state has an overall count of 9066 nodes.

In practice, Lhasa DB is able to reduce this amount of nodes up to 44,6 percent. Firstly, the prototypic implementation chooses a compact representation of relationships. That means, not every relationship in the sample also ends up in a de facto relationship node instance. This lowers the node quantity to 6070 units. Secondly, Lhasa DB uses unification to prevent identical values being stored, which cuts down the node number to finally 4040. Additionally, the overall edge count is also reduced while preserving the data integrity.

## 5.2. Measurements

The resource requirement of a database is essential for its application. That is the reason, why detailed measurements are performed and discussed in this section. The measuring problem is the complete composition of the database state along the sample data. Memory requirements are measured in kilobyte (KB) and the computing time in seconds (s). Additionally, the number of nodes and edges are counted.

The measurements are based on a comparison of overall nine different treatment scenarios. In the first scenario, the unification is completely disabled. The second scenario uses unification, but only for plain nodes and only in deferred mode. The third scenario uses instant as well as deferred unification exclusively for plain nodes. Subsequently,

the fourth scenario uses unification exclusively for structured types and so forth. Table 1 gives a complete overview.

**Table 1.** Measurement Scenarios

Acronym	Meaning
NU	no unification
PDU	exclusive deferred unification of plain nodes
PIDU	exclusive instant and deferred unification of plain nodes
SDU	exclusive deferred unification of structured nodes
ADU	exclusive deferred unification of aggregation nodes
AIDU	exclusive instant and deferred unification of aggregation nodes
MIDU	exclusive instant and deferred unification of multi nodes
CDU	complete unification (all nodes, only deferred)
CIDU	complete unification (all nodes, instant and deferred)

The measurements only deal with the constitution of the database state. The import of the sample data from a text file as well as the data preparation happens before. Based on the fact, that a profiler may tamper the test results, the loading time as well as the memory usage is taken manually by means of the process information. Every test series contains 100 values for every data point, for which the average as well as the standard deviation is calculated to estimate the sample quality. A measure is considered valid, if the standard deviation lies under 5 percent.

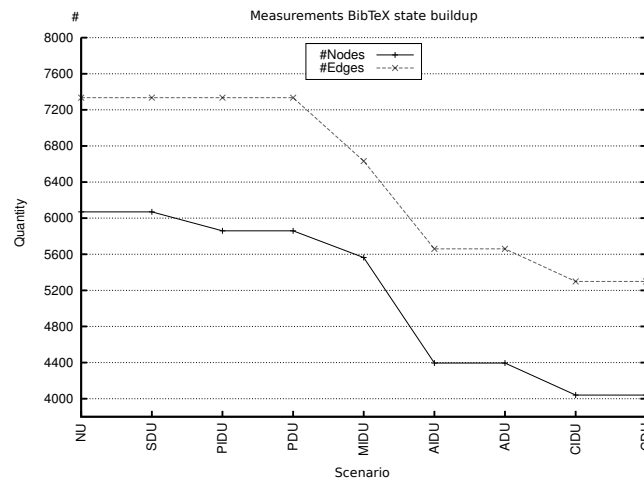
The measures are performed on a virtual Linux machine (VirtualBox 4.0.18 with Extension Pack on Ubuntu 12.10, GNU/Linux 3.5.0-17-generic x86\_64) with 512 MB main memory and a 64 bit Intel Core 2 Duo CPU (T9550, 6 MB Cache, 2.66 GHz, 1066 MHz FSB). The prototype is implemented in Ruby 2.0.0p0, which uses the YARV virtual machine. The Linux virtual machine is restarted after every pass to prevent side effects induced by cache contents. Additionally, the system workload is kept minimal on the host as well as on the guest system to not slow down the ruby execution speed.

Before discussing the results, one must be aware of the fact, that the memory and loading time results directly correlate with the nature of the sample data. So, all beneath statements cannot be generalized. Besides, there is no constraint checking done yet.

Figure 7 shows, that the number of nodes varies between 6070 and 4040 and the number of edges between 5299 and 7335. This difference can be explained by the different unification scenarios, which realize varying compression rates. The element count is laid on the left y axis. The node count is represented by a solid line and the edge count by a dashed one. The first scenario NU does no unification, so the state consists of the maximum node and edge quantity. The same applies to the second scenario SDU, because there are no equal author names and thus no unification is possible. The third scenario PIDU contains a reduction to 5860 nodes, what can be explained by the fact, that some authors have the same first or last name. Also the plain year annotation type can be unified in multiple cases. Based on the fact, that no structured nodes are unified, the edge count stays the same at 7335.

The fourth scenario PDU only differs in the time when the unification is done and thus establishes an equivalent state. The MDU scenario however shows a clear reduction of the node and edge quantity. The unification of multi nodes related to the `FirstName`, `LastName` as well as to the multi aggregation nodes saves a lot of space. All in all, the

node count is reduced to 5564 and the edge count to 6632. The same effect shows up in the following scenarios AIDU and ADU, where more aggregation nodes can be unified and thus the count downsizes again to 4395 nodes and 5560 edges. The last two scenarios CIDU and CU use the complete unification, which reduces the node number to finally 4040 and the edge count to 5299. Thus, the node compression rate lies at 33.4 and the edge compression factor at 27.9 percent.



**Figure 7.** Node and Edge Quantity Measurements

Figure 8 shows the required main memory as well as the loading time. The scenarios are sorted by the state loading time in ascending order. The memory usage is laid on the left y axis and is represented by a solid line. The loading time is laid on the right y axis and drawn as a dashed line. The main memory requirements vary between 9397.12 and 7901.52 KB, whereas the loading times differ roughly in a two tenth seconds range and reside between 1.1589 and 1.2910 seconds. Contrary to the above discussion, the distinction between instant and deferred unification plays in important role regarding the loading time.

The first scenario AIDU exhibits the least loading time with 1.1589 s. This scenario enables the instant unification of aggregation nodes in contrast to the ADU scenario (1.2025 s), where only deferred unification is active. So, in the latter scenario, the maximum of aggregation nodes are initially instantiated and then partly dropped, when unification occurs.

The second scenario PIDU only unifies plain nodes, what leads to a heavy increase of memory usage with 9255.76 KB. This also explains the raised loading time, because the maximum node count must be build for remaining node types. The same applies to the following scenario PDU. The fourth scenario NU neglects the unification at all and so the maximum number of nodes and edges must be established. The loading time increases to 1.1693 seconds and the memory usage reaches the maximum measured at 9397,12 KB. As the following scenario SDU virtually offers no unification candidates, the values resemble strongly with 1.1703 s and 9383 KB. The sixth scenario MDU marks the beginning of a significant change concerning loading time. In this scenario, the multi nodes are only unified in deferred mode. So the maximum of redundant multi nodes are

instantiated. On the one hand, this creates a memory overhead, because much more ruby objects must be initiated, and on the other hand, it causes additional computation time, because a lot of nodes must be reallocated respectively deleted during the unification process. This phenomenon can be also observed in the following example ADU.

The last but one scenario CIDU unifies all nodes types immediately, if possible. Against the prospects of the authors, the loading time increases to 1.2508 clearly. One possible explanation lies in the fact, that a lot of attribute index requests are performed, what annihilates the advantages of reduced node instantiations. The longest loading time (1.2910 s) in the last scenario CDU can be justified with the much higher reorganization costs of deferred unification. The last scenario also takes a bit more memory, which cannot be explained so far.

The results support the hypothesis, that the unification has a negative loading time impact. This effect gets even worse, if the unification happens deferred. In this case, a reduction of main memory requirements leads to a clearly increased loading time and vice versa.

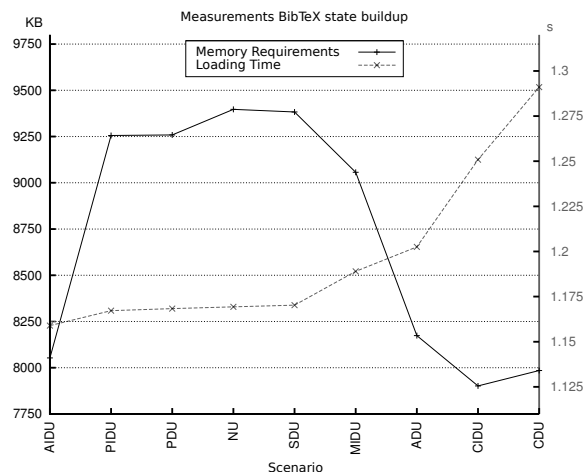


Figure 8. Time and Space Measurements

## 6. Conclusion

TEGeL complies with the requirements to be used for a conceptual and simultaneously for a logical and physical model, which simplifies the data modeling process. Furthermore, it also introduces concepts of object-oriented software engineering, like inheritance or interfaces. It is possible to express integrity constraints and the introduction of polymorphic aggregation leads to very flexible definition of entity type relations. Additionally, structured types, entity compositions and relationships of any kind can be modeled. TEGeL improves the semantic coherence, because every schema element has a unique name. Moreover, designers gain the possibility to reuse attributes. The direct expression of generalization, structured types, interfaces, aggregation, classes and clusters boosts semantics and eases the domain representation.

Lhasa DB uses TEGeL type graphs as logical schemata and establishes database states directly by deriving instance graphs. This procedure is described in detail above. Additionally, this paper explained the unification as a possibility to compress database states and to fasten index look-ups. We assume that unification generally leads to a very compact data representation, as it builds on the fact, that attribute values are discrete and often equal.

The results of the BibTeX experiment show, that a node takes up nearly 2 KB memory in average. This value must be seriously cut down to make Lhasa DB scalable. Current computer systems provide up to 2 TB of main memory<sup>2</sup> and could theoretically handle up to one billion nodes depending on the data quality. Larger memory requirements can be handled by distributed in-memory operation as it is discussed in [15]. Concerning performance, one must consider effective indexing methods as discussed in [13] and [14].

In addition to further implementation work, TEGeL may be extended to support *Role Based Access Control*. Information about roles and permitted operations can be attached to the membership edges realizing a very fine access control on the level of single attributes. A graph-based approach is introduced in [17]. When it comes to multi user access, a stable transaction concept must be elaborated. Additionally, the specific retrieval of data can be reduced to the induced subgraph isomorphism problem by formulating query conditions as search graphs. In this regard [29], [32], [35] as well as [34] can be considered.

Furthermore, the materialization process must be designed in detail and an efficient synchronization algorithm must be developed. In this context, one may think about additional data compression with *gzip*<sup>3</sup> or *value only unification* of nodes irrespective of their types. Looking at fast SSD devices for materialization can also be an interesting investigation area as the partial swapping of main memory data to fast hard disks provides an additional possibility to extend storage capacity.

## References

- [1] Abiteboul, S., Hull, R.: Ifo: a formal semantic database model. In: Proceedings of the 3rd ACM SIGACT-SIGMOD symposium on Principles of database systems. pp. 119–132. PODS '84, ACM, New York, NY, USA (1984), <http://doi.acm.org/10.1145/588011.588029>
- [2] Andries, M., Gemis, M., Paredaens, J., Thyssens, I., den Bussche, J.V.: Concepts for Graph-Oriented Object Manipulation. In: Pirotte, A., Delobel, C., Gottlob, G. (eds.) EDBT. Lecture Notes in Computer Science, vol. 580, pp. 21–38. Springer (1992)
- [3] Angles, R.: A Comparison of Current Graph Database Models. In: Kementsietsidis, A., Salles, M.A.V. (eds.) ICDE Workshops. pp. 171–177. IEEE Computer Society (2012)
- [4] Badia, A.: Extending Entity-Relationship Models with Higher-Order Operators. In: Ras, Z.W., Ohsuga, S. (eds.) ISMIS. Lecture Notes in Computer Science, vol. 1932, pp. 321–330. Springer (2000)
- [5] Barcia, R., Hambrick, G., Brown, K., Peterson, R., Bhogal, K.S.: Object Relational Impedance Mismatch. IBM Press, Westford (Massachusetts) (2008)
- [6] Böhm, R., Fuchs, E.: System-Entwicklung in der Wirtschaftsinformatik. vdf, Hochsch.-Verl. an der ETH, Zürich, 5 edn. (2002)
- [7] Chen, P.P.: The entity-relationship model - toward a unified view of data. ACM Trans. Database Syst. 1(1), 9–36 (1976)

---

<sup>2</sup>see Dell PowerEdge R910

<sup>3</sup>an open data compression tool



- [8] Codd, E.F.: Data Models in Database Management. In: Brodie, M.L., Zilles, S.N. (eds.) Workshop on Data Abstraction, Databases and Conceptual Modelling. vol. 11, pp. 112–114. ACM Press (1980)
- [9] Fowler, M.: Patterns of Enterprise Application Architecture. Pearson Education, Inc., Boston (2003)
- [10] Graves, M., Bergeman, E., Lawrence, C.: Graph database systems. *Engineering in Medicine and Biology Magazine, IEEE* 14(6), 737–745 (1995)
- [11] Gumm, D., Janneck, M., Langer, R., Simon, E.J.: Mensch - Technik - Ärger? Zur Beherrschbarkeit soziotechnischer Dynamik aus transdisziplinärer Sicht. Lit Verlag Dr. W. Hopf Berlin, Münster (2008)
- [12] Have, C.T., Jensen, L.J.: Are graph databases ready for bioinformatics? *Bioinformatics* 29(24), 3107–3108 (2013)
- [13] Islam, S., Fariha, A., Ahmed, C.F., Jeong, B.S.: EGDIM: evolving graph database indexing method. In: Lee, S.H., Hanzo, L., Ismail, R., Kim, D.S., Chung, M.Y., Lee, S.W. (eds.) ICUIIMC. p. 56. ACM (2012)
- [14] Jin, R., Ruan, N., Xiang, Y., Wang, H.: Path-tree: An efficient reachability indexing scheme for large directed graphs. *ACM Trans. Database Syst.* 36(1), 7 (2011)
- [15] Jouili, S., Reynaga, A.: imGraph: A Distributed In-Memory Graph Database. In: SocialCom. pp. 732–737 (2013)
- [16] Jouili, S., Vansteenbergh, V.: An Empirical Comparison of Graph Databases. In: SocialCom. pp. 708–715 (2013)
- [17] Koch, M., Mancini, L.V., Parisi-Presicce, F.: A graph-based formalism for rbac. *ACM Trans. Inf. Syst. Secur.* 5(3), 332–365 (2002)
- [18] Lehner, F., Wildner, S., Scholz, M.: *Wirtschaftsinformatik: Eine Einführung*. Carl Hanser Verlag, München · Wien, 2 edn. (2008)
- [19] Liu, Y., Vitolo, T.M.: Graph data warehouse: Steps to integrating graph databases into the traditional conceptual structure of a data warehouse. In: BigData Congress. pp. 433–434. IEEE (2013)
- [20] Macko, P., Margo, D.W., Seltzer, M.I.: Performance introspection of graph databases. In: Kat, R.I., Baker, M., Toledo, S. (eds.) SYSTOR. p. 18. ACM (2013)
- [21] Mamadolimov, A.: Search Algorithms for Conceptual Graph Databases. CoRR abs/1207.2837 (2012)
- [22] Møller, A., Schwartzbach, M.I.: *An introduction to XML and web technologies*. Addison-Wesley (2006)
- [23] Neubauer, W., Rudow, B.: *Trends in der Automobilindustrie*. Oldenburg Wissenschaftsverlag GmbH, München (2012)
- [24] Pluciennik, T., Pluciennik-Psota, E.: Using Graph Database in Spatial Data Generation. In: Gruca, A., Czachórski, T., Kozielski, S. (eds.) ICMMI. *Advances in Intelligent Systems and Computing*, vol. 242, pp. 643–650. Springer (2013)
- [25] Poulouvassilis, A.: Database research challenges and opportunities of big graph data. In: Gottlob, G., Grasso, G., Olteanu, D., Schallhart, C. (eds.) BNCOD. *Lecture Notes in Computer Science*, vol. 7968, pp. 29–32. Springer (2013)
- [26] Pressman, R.S.: *Software Engineering: A Practitioner’s Approach*. McGraw-Hill, New York, 7 edn. (2010)
- [27] Rupp, C.: *Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis*. Carl Hanser Verlag, München, 5 edn. (2009)
- [28] Samiullah, M., Ahmed, C.F., Nishi, M.A., Fariha, A., Abdullah, S.M., Islam, M.R.: Correlation Mining in Graph Databases with a New Measure. In: Ishikawa, Y., Li, J., Wang, W., Zhang, R., Zhang, W. (eds.) APWeb. *Lecture Notes in Computer Science*, vol. 7808, pp. 88–95. Springer (2013)
- [29] Sundaram, G., Skiena, S.S.: Recognizing small subgraphs. *Networks* 25, 183–191 (1995)
- [30] Thalheim, B.: Extended Entity-Relationship Model. In: Liu, L., Özsu, M.T. (eds.) *Encyclopedia of Database Systems*, pp. 1083–1091. Springer US (2009)
- [31] Thomas, R.: Introduction to the Unified Modeling Language. In: TOOLS (25). p. 354. IEEE Computer Society (1997)
- [32] Ullmann, J.R.: An algorithm for subgraph isomorphism. *J. ACM* 23(1), 31–42 (Jan 1976), <http://doi.acm.org/10.1145/321921.321925>
- [33] Wallmüller, E.: *Software-Qualitätsmanagement*. Carl Hanser Verlag München, München · Wien, 2 edn. (2001)
- [34] Yuan, Y., Wang, G., Chen, L., Wang, H.: Efficient subgraph similarity search on large probabilistic graph databases. CoRR abs/1205.6692 (2012)
- [35] Zheng, W., Zou, L., Lian, X., Wang, D., Zhao, D.: Graph similarity search with edit distance constraint in large graph databases. In: He, Q., Iyengar, A., Nejdil, W., Pei, J., Rastogi, R. (eds.) CIKM. pp. 1595–1600. ACM (2013)

# A Dengue Location-Contraction Risk Calculation Method for Analyzing Disease-Spread

Wahjoe T Sesulihatien<sup>1,2)</sup>, Yasushi Kiyoki<sup>2)</sup>

<sup>1)</sup> *Electronics Engineering Polytechnic Institute of Surabaya, Indonesia*

<sup>2)</sup> *Keio University, Shonan Fujisawa Campus, Japan*

**Abstract.** *Dengue fever is the fastest spreading communicable disease in the world. The virus has been increasing its geographic reach, partly due to increased urbanization and partly due to climate change. From the viewpoint of human movement, population density the most suspected factor in spreading, but some results nowadays show that the relationship between population density and dengue fever is unclear. This paper presents a new approach in measuring human involvement by modelling contagious places. The proposed system includes (1) statistical analysis about correlations between contagious places and Dengue fever cases, (2) a multi-layer weighting method for determining the weight of each cell (3) a ranking and classification method for places' human-mingling, and (4) building a risk-map of contagious places as a control method in Dengue spreading. The result is new dimension to measure vulnerability of land use concerning with pattern of human moving. Our new approach is advantageous for effecting monitoring in change of public facilities, in comparison to the approach based on the population density. It is more useful in urban planning due to priority in evacuation, and control dengue based on places that attract people.*

**Keywords.** *Dengue, weighting method, risk control, contagious place, population density*

## Introduction

Dengue fever is a painful, debilitating mosquito-borne disease caused by one of four closely related dengue viruses. It is transmitted by the bite of an infected *Aedes* mosquito. The mosquito becomes infected when it bites a person with dengue virus present in their blood. It is not transmitted directly from person-to-person. Until now, more than 100 million cases of dengue fever occur worldwide. Most of these are in tropical areas of the world, with the greatest risk occurring in: The Indian subcontinent, Southeast Asia, Southern China, Taiwan, The Pacific Islands, The Caribbean (except Cuba and the Cayman Islands), Mexico, Africa, Central and South America (except Chile, Paraguay, and Argentina), southern United States, and southern Australia [1]. In Indonesia, dengue cases increase yearly in almost all regions—especially in Bali—followed by Central Sulawesi, the Riau Islands, Jakarta, Jambi, Aceh, Riau, West Sumatra, North Sumatra and Bengkulu. [2] The virus has been spreading in geographic reach, partly due to increasing urbanization [3]-[7] and partly due to climate change [8]-[12].

In earlier research about dengue, population density is considered as representative of human movement. Theoretically, the higher population density the higher possibility of disease. This has been shown in some studies done in areas like Kuala Lumpur [13], Jeddah [14], Barbados, Brazil, and Thailand [15]. In reality, individuals vary considerably in the frequency, distance, and nature of their movements [16]. Research in Vietnam reported that cases of dengue occurred in low-to-moderate population densities [17], while in Cambodia [18] and Peru [19] there is no significant relationship between population density and cases of dengue. The contrast leads to explore another variable that more understandable to recognize human-involvement in spreading mechanism.

Some studies in dengue are now focused on patterns of human movement. Research in Penang [20] shows that case-distribution trends depend on whether an area had previous dengue cases. In Peru [21], transmission appears to be shaped by social connections, because routine movements among people in the same places. In Cambodia [18], along the busiest national road of the country, synchronously occurring epidemics over large geographic areas are common, whereas rural areas experience travelling waves of outbreaks. While in UK [22], Leon Danon found that routine, daily commuter-type movements lead to a slower epidemic spread compared to movements with random destinations. Even though some researchers agree that movement of humans is a factor in dengue spread, nothing is mentioned clearly about variable or factor that represents human moving and how to measure this involvement.

To examine the problem, this paper presents a novel analysis of contagious-place risk-ranking that explicitly represents regular and routine movements. We calculate the weighting of multi contagious place types based on data of cases in 2011, and use this model to examine the relative impact of transmission as an effect of contact among people. This paper also identify ranking of contagious places overlay's and population density, and compare both to determine which one is better to represent human movement. To this end, the paper is organized as follows. In the Proposed Method section we introduce a model and describe its accompanying data. The Results and Discussion section describes the methodology by which we compare the impact weight of contagious-places and present the main results from these analyses. Finally the implications of this study will be discussed in the Conclusion section.

## 1. Proposed Method

The new approach that we propose in dengue spreading is developed based on human to human contact by using agent-based models (ABM) where each individual is explicitly represented and characterized. Agent-based models allow interaction among individual actors or "agents", an environment, and a set of rules [23]. Actions in ABM take place through the agents: simple, self-contained programs that collect information from their surroundings which are applied to a pre-determined scenario.

In case of dengue spreading, the system explores the interactions among agents in contagious-places with weather as a variable. There are three main aspects to this simulation: the human-context, the mosquito-context and interaction of humans and mosquitos. The overall system is shown in Figure 1.

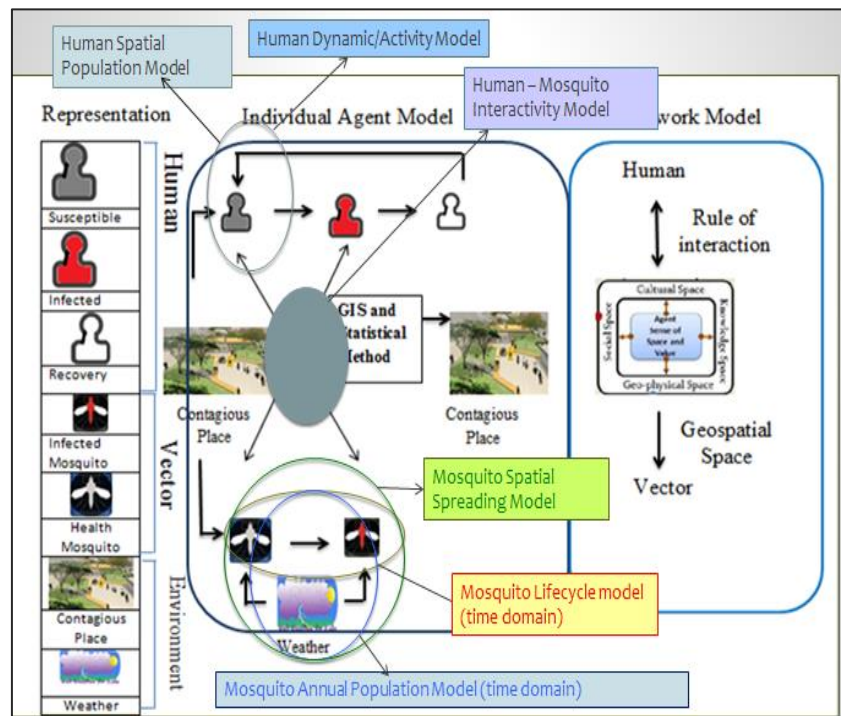


Figure 1. Overall System

The overall system integrates the human-to-human model and mosquito-to-human model. As we know, transmission among people (called hosts) happens via mosquitos (called the vector). Human-to-human modelling consists of a human-spatial-population model and a human-dynamic/activity-model. The human-spatial-population model in this study is based on spatial statistical analysis, while human-dynamic/activity is represented by risk-ranking of contagious-places. The mosquito-aspect involves spatial spreading, life cycle, and annual mosquito population. These are utilized within the human-mosquito interactivity model. The objective of this paper is to represent dynamic human activity by using a risk-map of contagious-places in a

defined area. It means that firstly, the evaluation criteria have to be chosen for the various risk-places. Secondly, methods have to be identified in order to assess these risk-criteria in a spatially-differentiated way, such as risk maps being created for each of the chosen criteria. And thirdly, these different risk maps are aggregated by means of appropriate multi-criteria decision-rules in order to get an overall risk assessment and final mapping result. The schema of the risk map method is illustrated in Figure 2.

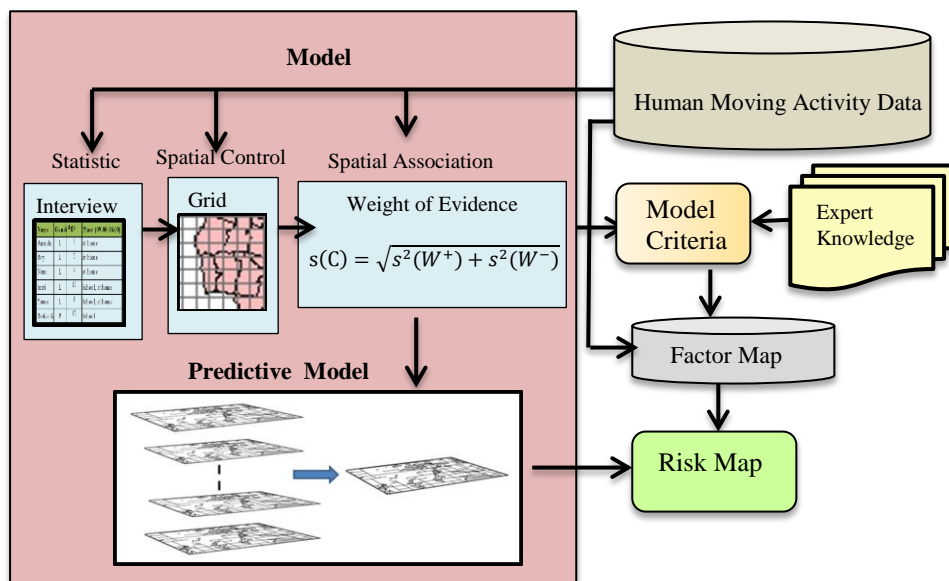


Figure 2. Risk Map Method

Statistical analysis is applied to analyze variable that symbolize disease transmission. All variables are then mapped and controlled in small area by grid. A weight-of-evidence method is utilized to extract values from spatial-associations. The result is a risk-map of Surabaya. Indeed, spatial-association analysis completing spatial distribution analysis because its results are quantitative [24]. The results of these analyses are combined with experts' opinions to define a risk model of human movement. The evidential features are transformed to factor maps, which are utilized as input data for prediction models. The factor-maps are inputs for driving knowledge prediction models. The results are then integrated into evidential-map layers for producing a prospective map. The prospective map is then classified into importance classes. Furthermore, the prediction-rate of each importance class is estimated, and the output is a risk-ranking.

### 1.1 Data Preparation

Area study is divided in a small control area namely grid. In this case, a grid area of 1 km<sup>2</sup> is selected for the following reasons: (1) This is common standard size for control in population such as in Japan, US and (2) Kelurahan as the smallest administrative district in Surabaya has a dimension mostly 0.8 km<sup>2</sup> to 1.5 km<sup>2</sup>, and (3)

The flight range of *Aedes aegypti* is 100 meters, and a the maximum flight are 400m. [25].

Population data is collected as secondary data from Surabaya City in terms of population number in each Kelurahan. Therefore, one grid probably consists of more than one area. For this case, datasets were interpolated by statistical equation [26 ]:

$$D = \frac{\sum_{i=1}^n D_i x A_i}{n} \dots\dots\dots(1)$$

Where D is the population density in a specific grid that performed by i Kelurahan, A<sub>i</sub> is percentage of area kelurahan to grid area (1 km<sup>2</sup>) and n is number of grid area

Then, all data from three maps: map of WOE, map of population density and map of dengue case as validation, are classified into 10 levels. For map of dengue case, classification is done base on number of case in every grid : class 1 to 7 is represent 0 to 7 case while class 8 represent 8-9 case and class 10 represent 10 or more case.

**1.2 Risk Map**

In this research, the goal is to determine the vulnerability-ranking of a contagious-place performed by risk map. The Weight of Evidence approach is applied to produce a predictive model of dengue case occurrences. The GIS-based susceptibility assessment comprises the following main processing steps [27]: (1) compiling place of contact’s inventory, (2) separating inventory into a modelling and validation set, (3) deriving dengue spreading control from source data by GIS analysis, (4) calculating weights for each controlling factor by using the modelling set of spreading, (5) generate multiclass generalization evidences based on the cumulative weighting, (6) calculating posterior probability map (i.e., combination of the controlling factors to predict potential dengue case occurrences), and (7) validating the model using the validation set of the inventory.

The first through third step includes collecting data (contagious place, dengue case, population density), digitizing the data, layering it in separate maps, and assembly suspected contagious place with dengue case.

The fourth step is determining the method for weighted spatial association. In this study, Weights of Evidence (WOE) is used. WOE is the spatial association analysis of a given region D by weighting inside and outside of the given region [28] The weighting yields (1) W<sup>+</sup> is weights within the region (D<sup>P</sup>), and (2) W<sub>-</sub> is weights outside the region (D<sup>N</sup>) and T = D<sup>P</sup> + D<sup>A</sup>; T is the total study area. Then weight positive and negative is written as :

$$W^+ = \ln \left( \frac{P(B_i|s)}{P(B_i^+)} \right) \dots\dots\dots(2)$$

$$W^- = \ln \left( \frac{P(B_i|s)}{P(B_i^-)} \right) \dots\dots\dots(3)$$

W<sup>+</sup>>0 implies positive spatial association; and W<sub>-</sub>> 0 implies negative spatial association; while W<sup>+</sup>= W<sub>-</sub>= 0 implies no spatial association. For example we want to get WOE of area where has 1 school. In this case s means area evidence (area victim due to class i), B<sub>i</sub> = class i(school), B<sub>i</sub><sup>+</sup>= evidence of positive (1 school is with dengue case), B<sub>i</sub><sup>-</sup>= evidence of negative (1 school without dengue case).

The differences between  $W^+$  and  $W^-$  indicate contrast relation in both spatial association parameters. Contrast is denoted as

$$C = W_i^+ - W_i^- \dots\dots\dots (4)$$

If  $C > 0$ , spatial association is positive, conversely  $C < 0$ , and in the case of  $C = 0$  there is no spatial association. Furthermore, the maximum of  $C$  can determine the optimum cutoff weight from features.

The uncertainty of the weights and contrast can contribute to the interpretation of the contrast; it is denoted as :

$$s(C) = \sqrt{s^2(W^+) + s^2(W^-)} \dots\dots\dots (5)$$

The uncertainty of the weights and contrast would be large if the number of resource sought points is small in the study area and causing meaningless or unreal result. For solving this problem, the standardized measure of  $C$  is usually calculated as the ratio of  $C$  to its standard deviation,  $C/s(C)$ , which shows the significance of the spatial association [28.]

$$stud(C) = \frac{C}{s(C)} \dots\dots\dots (6)$$

Each cell's weighting is determined by comparing  $stud$  of several classes. For this purpose, expert-opinion is utilized to determine close rankings between contagious places and dengue cases [29].

The fifth step is generating multiclass evidence based on the results of several layers. Each evidence-map layer converting three criteria: cutoff distance, weight and inter-layers scores to factor map. In the multi-class index overlay model, the classes on each  $i$  evidential map layer input resulting different scores by Index Overlay equation [30] :

$$S = \frac{\sum_i^n w_i S_{ij}}{\sum_i^n w_i} \dots\dots\dots (7)$$

where :  $w_i$  is the weight of the  $i$ -th map,

$S$  is index overlay system

$S_{ij}$  denotes the index overlay for every class: value is 1 for presence or 0 for absence of the binary condition.

The output score is between 0 (implying extremely safe) to 1 (implying high vulnerability). The result is a risk map showing regions from low to high risk.

The last procedure is the validation method. In this case, the risk map of prospective map based on contagious map and risk map based on population density will compared with risk map based on dengue case.

## 2. Experimental Result and Discussion

### 2.1 Study Area

Symptomatic Dengue case data was obtained for 2011 from municipal health departments in Surabaya, Indonesia. Case-occurrence data includes the address of the victim. There were 1088 total cases reported from January to December 2011. These cases were scattered across 166 Kelurahan (smallest administrative district) as shown in Figure 3.

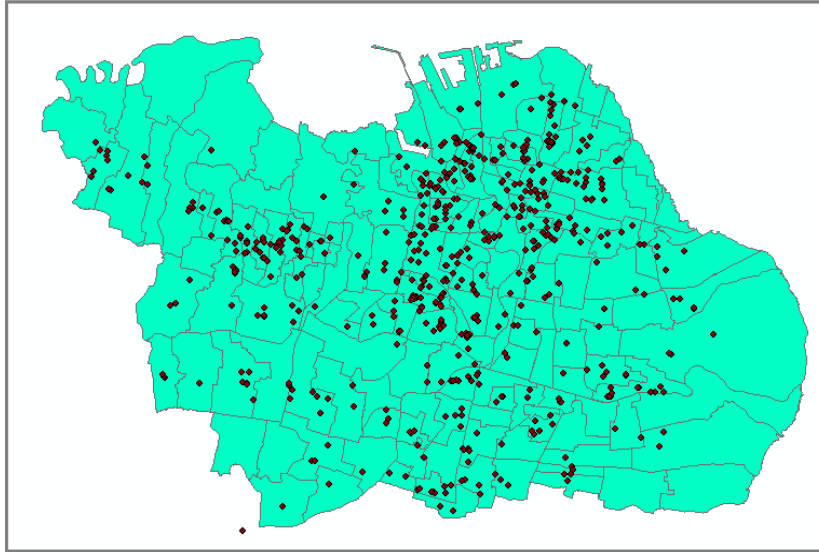


Figure 3 Dengue Case

Fine-resolution geospatial data characterize pattern of routine moving variation in Surabaya for one year. Estimation of potential risk areas for dengue case occurrences based on indirect and landscape-scale measure. It is correlated to transmission factors among disease-vector-host interactions that make up the etiology of this disease. In particular, it will focused on correlation among several suspected contagious place types: point, line and polygon in table 3

Table3. Suspected Contagious Place

Layer of Place	Type
Education Facilities	Point
Industrial and Warehouse	Polygon
Traditional Market and Supermarket	Point
Commerce and business	Line
Residential	Polygon
Mall	Point

The suspected contagious-place is selected based on interviews and correlation with dengue cases by using the Pearson Correlation method [31]. Each data-point is overlaid in a grid to determine area for control. The result is shown in Figure 4.



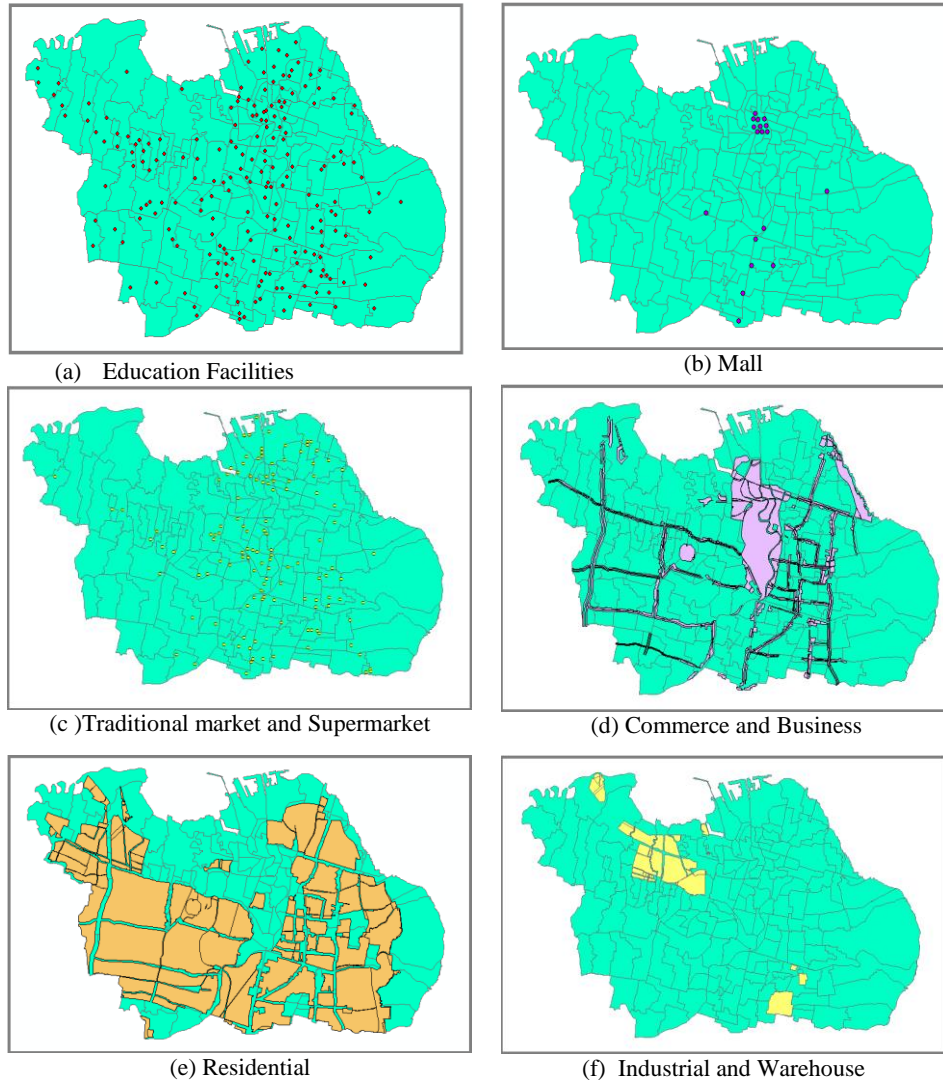


Figure 4. Suspected Contagious Areas

## 2.2 Risk Map

The Weight of Evidence (WOE) method is applied to assess the vulnerability distribution of dengue. WOE is a general-purpose method that generates predictions from multi-layered sets of evident occurrence based on a probabilistic framework. In this calculation, the probability of occurrence is applied for evaluating the overall weight. The weight of each grid expresses the probability of dengue transmission as a function of its human activity. A high value in a particular grid cell indicates suitable conditions for transmission; or in this case high probability for DF case-occurrences.

Figure 5 shows the prevalence of some contagious-places that are suspected dengue transmission places.

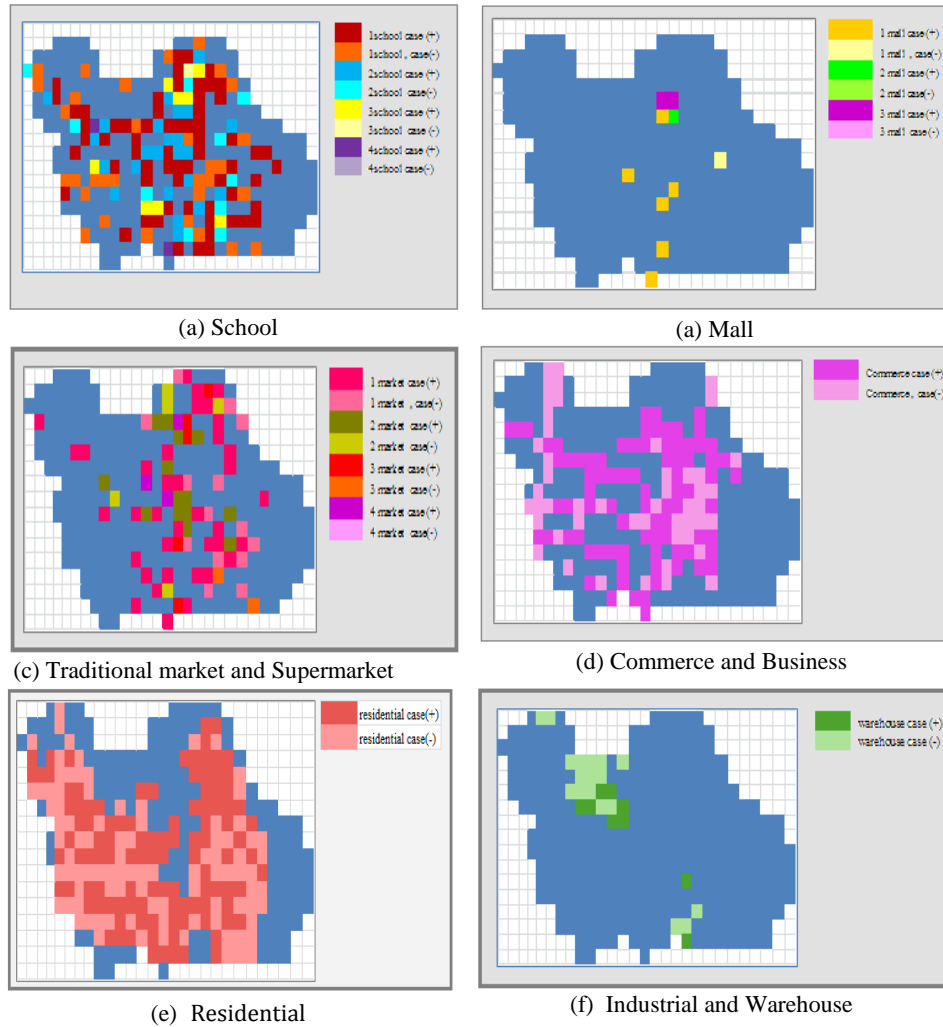


Figure 5. Prevalence of Contagious Place

Figure 5 shows the presence or absence of evidence variable (called class) in dengue case. The area in question is shown on the map in blue. Presence of evidence is shown in dark color and absence in light color. Presence means in a given cell, there are variable with dengue case while absence means there are variable without dengue case. For example, Figure 5(a) shows schools as a variable: brown denotes 1 school, sky-blue denotes 2 schools, yellow denotes 3 schools and purple denotes 4 schools. Dark brown means there is 1 school in this area with a reported dengue case while light brown indicates 1 school in this area without a dengue case. Another figure in Figure 5 illustrates another variable of suspected contagious area.

In the case of categorical data, weights are calculated for each class of the variable separately. Variations of weighting interpret importance of classes on dengue spread proneness. However, for the final calculation of the posterior probability, the datasets are generalized in order to reduce the number of classes. After generalization, the weights are calculated categorically for each class separately. The result is shown in Table 4.

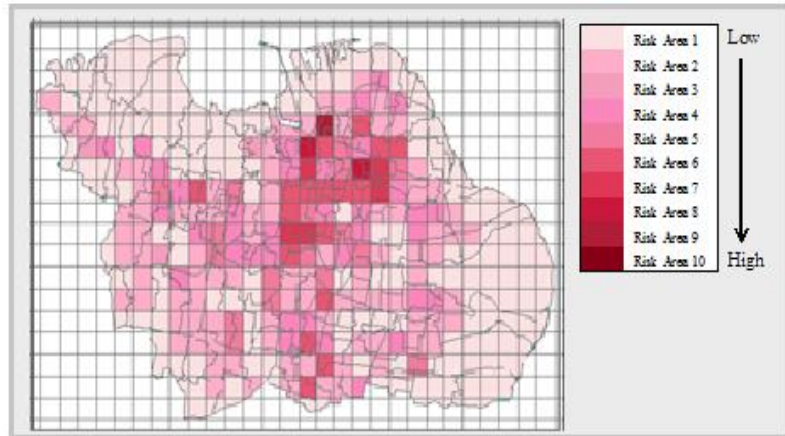
Table 4 Variation of weights of evidence

Variable	number	W+	W-	C	stud(c)
School	1	-0.7755908	-1.5004867	0.724895879	22.471772
	2	-1.7989797	-2.6318888	0.832909123	8.3290912
	3	-2.8550324	-4.9344739	2.079441542	2.0794415
	4	-3.8358616	-4.9344739	1.098612289	1.0986123
Industry and Warehouse		-1.0986123	-0.4054651	-0.693147181	-12.476649
Trad market and Supermarket	1	-0.7731899	-1.7917595	1.018569581	13.241405
	2	-1.7176515	-2.5649494	0.84729786	5.0837872
	3	-2.9704145	-3.6635616	0.693147181	1.3862944
	4	-3.2580965	-4.3567088	1.098612289	1.0986123
Mall	1	-0.5877867	-2.1972246	1.609437912	1.6094379
	2	-1.0986123	-2.1972246	0.693147181	0.6931472
	3	-1.5040774	-2.1972246	0.693147181	0.6931472
Business and Commerce		-0.4563567	-1.0039963	0.547639597	26.286701
Residential		-0.3884868	-1.133459	0.744972248	70.027391

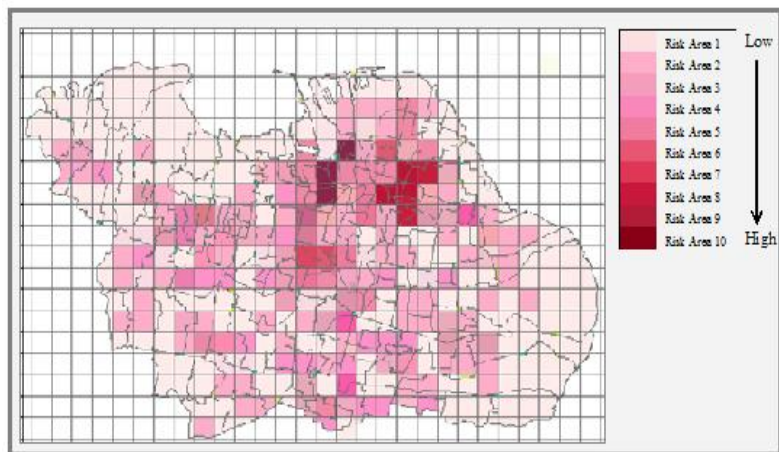
Final categorical weighting is calculating multi-class weights. In this case, expert opinion plays an important role in judging the importance among variables, such as determining the hierarchy of importance for every class, and determining comparisons over the entire range of variables. A numerical weight or priority is derived for each element of the hierarchy, allowing diverse and often incommensurable elements to be compared to one another in a rational and consistent way. Then, the weight of every cell is calculated by Equation 7. The result is a predictive map of the risk area.

### 2.3 Discussion

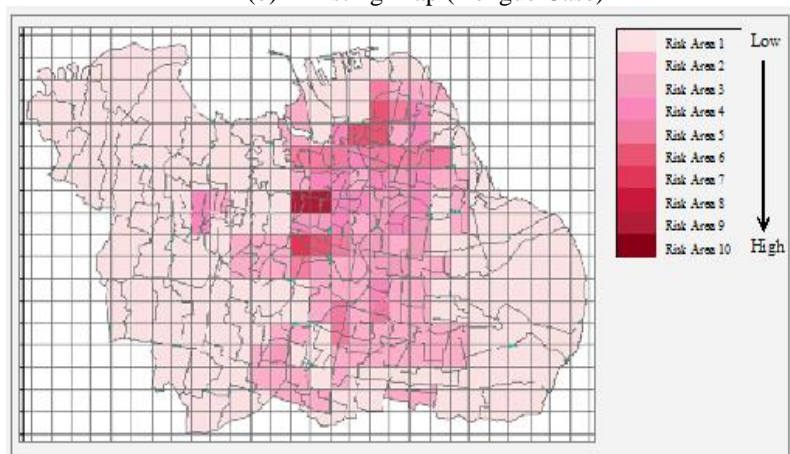
The objective of this study is to provide prevailing vulnerabilities to dengue fever in Surabaya, Indonesia by using a composite vulnerability index. The index builds on a set of underlying human movement indicators. In this case, population density and place of contact were compared to determine which one is better as a human-contributed factor in dengue spread prediction. For that reason, three maps will be compared: predictive map of contagious places as a proposed method, population density map as a representation of existing methods and a dengue-case incident report map as a real-data control.



(a) Proposed Map (Contagious Place)



(b) Existing Map (Dengue Case)



(c) Population Density Map

Figure 6 Comparison between Proposed, Existing and Population Density Maps

The areas being studied were divided into 328 cells that denote a total area of 328 km<sup>2</sup>. For predictive map, result of WEO is categorized by 10 risk area based on result of equation (7). The smallest population density is 9 and the highest is 2748. While the smallest case is 0 and the highest is 13 in every cell. Total case is 1088. This data is categorized by 10 risk area also.

It is difficult to count the comparable error of both map (predictive and population density) to dengue case map because in this case compare every cell is not just to justify this cell “same” or “not” but also to investigate level of differences. Every cell in both the predictive map and population density map is compared with the existing control-map.

For example, if a dengue-case appears in a cell with a level of 10, the same cell in the population-density map shows a level of 8, and the same cell in the predictive map indicates a level of 7, then the error in the population density map is 2, while the error in the predictive map is 3. Error in this case represents how much variation or dispersion exists compared to the real value. Therefore, the total error of each map is calculated by utilizing the standard deviation method. The result is shown in Table 5.

Table 5. Error and Deviation

No	(XPD-X <sub>ex</sub> )	N1	(XP-X <sub>ex</sub> )	N2	SDPD	SDP
1	0	171	0	239	1.506085	0.669456
2	1	88	1	78		
3	2	42	2	6		
4	3	11	3	5		
5	4	9	4	0		
6	5	3	5	0		
7	6	2	6	0		
8	7	2	7	0		
9	8	0	8	0		
10	9	0	9	0		

XPD, X and X<sub>ex</sub> indicate grid level of predictive map, population density map and existing map respectively. N1 and N2 correspondingly to number of grid that comprise (XPD-X<sub>ex</sub>) and (XP-X<sub>ex</sub>), while (XPD-X<sub>ex</sub>) is level differences of predictive map to existing map, and (XP-X<sub>ex</sub>) is level differences of population density map to existing map. Standard deviation (SD) expresses level of fault, SDPD is standard deviation of population density map and SDP is standard deviation of contagious place map as a proposed map.

By statistical analysis, it is clear that the total error of our proposed map is better than the Population Density map. It means that it is possible to predict the risk areas without relying on information about the population density, or the occurrence of dengue cases. The spatial relationships between occurrences of dengue cases correlated with human activity related to the human to human disease transmission has been completely integrated. It is sufficient to identify the risk areas for the whole study area. The risk areas are presented by developing a map that showing the place of contact in

the study area. This result is confirmed by several papers about population density in urbanization [2]-[7] and population density in “well mixed human populations”. [13]-[15]. All paper state about existence of public facilities such as business area, school, cemetery, etc, besides population density, even though level of their influence in risk is not mentioned clearly.

The new finding changes the meaning of Tobler’s first law of geography (1970) “everything is related to everything else, but near thing is more related than distant things”. In case of communicable disease, social dynamic is more important than distance. As an impact, spatial association method such as Mooran’s Index that correlate a signal among nearby locations in space should be consider long distance association. It also changes the method to determine outbreak. The classical outbreak based on global and local spatial associations such as LISA (Local indicators of spatial association) to evaluate the clustering in those individual units by calculating Local Moran’s I for each spatial unit and evaluating the statistical significance for each index. This method cannot explain why outbreak goes to long distance at almost the same time. Therefore, an index of vulnerability should be added during global and local spatial analysis [32]. In our study, this index of vulnerability is a risk level of public meeting-places.

Our new approach provides useful information to health authorities, and could assist in focusing and implementing prevention and control measures. Recently in Indonesia, dengue control measures are carried out randomly based on areas where victims are located, in order to avoid further spreading. But this is not effective, because people move dynamically, and thus have the opportunity to transmit dengue outside their area. Monitoring a number of public facilities is easier and more effective than monitoring a number of people in certain area. It is also useful for city planning with regards to public health. When a city grows large, governments should consider about communicable-disease evacuation when outbreak comes such as planning in hospital placement and managing disease control system based on dynamic city model.

In spite of the benefits of our calculated risk map, there are some limitations in its prediction capabilities. In this study, we assume that people go to certain places during business hours, so their movement is routine walk. Another limitation of this study is that contact is assumed to happen within ‘well-mixed’ human population where each individual is equally likely to encounter every other individual and every mosquito. In fact, some people move randomly during business hours and have different possibility to come into contact with mosquitos. It is alleged as uncertain factor in risk map [32]. Therefore, this model is more suitable for cities with certain patterns of movement, like metropolitan areas, rather than tourist areas where people diverse in immunity and activity.

### 3. Conclusion and Future Work

An important finding of this work is that the presence of contagious places could represent human involving in spreading of dengue fever better than population density. This finding simplifies the control of dengue spreading in at least two ways: (1) Monitoring of public facilities is easier to realize than monitoring of population density. This is because population density may increase and decrease dynamically, so it is difficult to include in a spreading model, and (2) it is more useful in urban planning due to priority in evacuation, and control dengue based on places that attract people.

Furthermore, this study provides a new dimension to measure vulnerability of land use concerning with pattern of human moving. Studies from an epidemiological are now needed to ascertain in greater detail how the social structure of human movement is quantitatively related to the risk of dengue infection in light of the inherent variations among different social groups. Our next research will be focused on human-mosquito transmission, and new scenarios of spreading based on human-mosquito interactions.

### Acknowledgments

This research was supported by Ministry of Education Indonesia scheme INTERNATIONAL RESEARCH COLLABORATION between EEPIS and KEIO University. The authors thanks the Ministry of Health district Surabaya for providing dengue case data.

### References

- [1] [www.who.int/csr/disease/dengue](http://www.who.int/csr/disease/dengue)
- [2] Departemen Kesehatan, 2010, Buletin Jendela Epidemiologi, Volume 2 tahun 2010, <http://www.depkes.go.id/downloads/publikasi/buletin/BULETIN%20DBD.pdf>
- [3] Derek A. T. Cummings, 2011, et all The Impact of the Demographic Transition on Dengue in Thailand: Insights from a Statistical Analysis and Mathematical Modeling [www.plosmedicine.org](http://www.plosmedicine.org) 1 September 2009 Volume 6, Issue 9, e1000139
- [4] DH Barmak et all, 2011, Dengue Epidemic and Human Mobility *PHys Rev E Stat Nonlin Soft Matter Phys*, 2011 Jul :84
- [5] Mathieu Andraud, Niehl Hens, Christiaan Marais, Phillipe Beutel, 2012, Dynamic Epidemiological Models for Dengue Transmission: Systematic Review of Structural Approaches, *PLoS ONE*7(11): e49085. doi:10.1371 /journal.pone.0049085
- [6] Abdullah G. Alzahrania, Mohammad A. Al Mazroaa, Abdullah M. Alrabeahb, Adel M. Ibrahimc, Ali H. Mokdadd and Ziad A. Memishe, f 2011, Geographical distribution and spatio-temporal patterns of dengue cases in Jeddah Governorate from 2006–2008, *Transactions RSTMH*, 10.1093/trstmh/trs01
- [7] Derek A. T. Cummings, 2011, et all The Impact of the Demographic Transition on Dengue in Thailand: Insights from a Statistical Analysis and Mathematical Modeling
- [8] Rachel Lowe et all, 2010, Spatio-temporal modelling of climate-sensitive disease risk: Towards an early warning system for dengue in Brazil, *Computer and Geoscience*
- [9] Yien Ling Hii, 2011, Forecast of Dengue Incidence Using Temperature and Rainfall, <http://www.plosntds.org/article/info%3Adoi%2F10.1371%2Fjournal.pntd.0001908>
- [10] Wiwiek Setya Winahju,, Adatul Mukarromah, Modeling Dengue Cases Using Poisson INAR, *Procedia Engineering* 50 ( 2012 ) 837
- [11] Szu-Chieh Chen, , Meng-Huan Hsieh, 2012, Modeling the transmission dynamics of dengue fever: Implications of temperature effects, *Science of the Total Environment* 431 (2012) 385–391

- [12] Zha Zhaoxia Wanga\*, Hoong Maeng Chana, Martin L. Hibberdb, Gary Kee Khoon, 2012, Delayed Effects of Climate Variables on Incidence of Dengue in Singapore during 2000-2010, *APCBEE Procedia* 1 ( 2012 ) 22 – 26
- [13] Hafiz Hassan<sup>1</sup>, Shamarina Shohaimi<sup>2</sup>, Nor R. Hashim<sup>1</sup>, 2012, Risk mapping of dengue in Selangor and Kuala Lumpur, Malaysia, *Geospatial Health* 7(1), 2012, pp. 21-25]
- [14] Abdulatif Alharty, 2009, Role of GIS in Dengue Control Management Strategy at Jeddah Municipality, [www.saudis.org/FCLFiles/File/ 33\\_E\\_AbdullatifAlharty\\_KSA.pdf](http://www.saudis.org/FCLFiles/File/33_E_AbdullatifAlharty_KSA.pdf)
- [15] Daniel Parker ,Darryl Holman ,2012, Event history analysis of dengue fever epidemic and inter-epidemic spells in Barbados, Brazil, and Thailand, *International Journal of Infectious Diseases* 16 (2012) e793–e798
- [16] Gonzalo M Vazquez-Prokopec ,2011, et al Quantifying the Spatial Dimension of Dengue Virus Epidemic Spread within a Tropical Urban Environment, , [http://www.plosntds.org rticle/info%3Adoi%2F10.1371%2Fjournal.pntd.0000920](http://www.plosntds.org/article/info%3Adoi%2F10.1371%2Fjournal.pntd.0000920), 2011
- [17] Wolf-Peter Schmidt Motoi Suzuki,Vu Dinh Thiem, Richard G. White,Ataru Tsuzuki,Lay-Myint Yoshida,Hideki Yanai,Ubydul Haque,Le Huu Tho,Dang Duc Anh, Koya Ariyoshi , 2011, Population Density, Water Supply, and the Risk of Dengue Fever in Vietnam: Cohort Study and Spatial Analysis, *PLoS Medicine* | August 2011 | Volume 8 | Issue 8 | e1001082
- [18] Magali Teurlei et al,2012, Can Human Movement Explain Heterogenous Propagation of Dengue Fever in Cambodia?, [www.plosntds.org/article /info%3Adoi%2F10.1371%2Fjournal.pntd.0001957](http://www.plosntds.org/article/info%3Adoi%2F10.1371%2Fjournal.pntd.0001957)
- [19] Steven T Stoddart et al, 2012, House to House Human Movement Drive Dengue Virus Transmission, <http://www.pnas.org>
- [20] C.D. Nazri , Hashim A. , Rodziah I, Abu Hassan, A. Abu Yazid ,2013, Utilization of Geoinformation Tools for Dengue Control Management Strategy: A Case Study in Seberang Prai, Penang Malaysia, *International Journal of Remote Sensing Applications* Volume 3 Issue 1, March 2013
- [21] Robert C. Reiner Jr, Steven T. Stoddard, Thomas W. Scott, 2014, Socially structured human movement shapes dengue transmission despite the diffusive effect of mosquito dispersal, *Epidemic*, Vol 6 March 2014, pages 30-36
- [22] Leon Danon, Thomas House, Matt J. Keeling, 2009, The role of routine versus random movements on the spread of disease in Great Britain, *Epidemics* 1 (2009) 250–258
- [23] Liliana Perez, Suzana Dragicevic, 2009, An agent-based approach for modeling dynamics of contagious disease spread, *International Journal of Health Geographics*
- [24] Majid Kiavar Moghaddama, Farhad Samadzadegana, Younes Noorollahib, Mohammad Ali Sharifia, Ryuichi Itoica, 2014, Spatial analysis and multi-criteria decision making for regional-scale geothermal favorability map, *Geothermics* 50 (2014) 189– 201
- [25] Surabaya Dalam Angka 2011, [www.surabaya.go.id](http://www.surabaya.go.id)
- [26] I.N. Gregory, 2002, The accuracy of area interpolation techniques: standardising 19th and 20th century census data to allow long-term comparisons, *Computers, Environment and Urban Systems*, Volume 26, Issue 4, July 2002, Pages 293-314



- [27] Bettina Neuhäuser , Bodo Damm , Birgit Terhorst, 2012, GIS-based assessment of landslide susceptibility on the base of the Weights-of-Evidence model, *Landslides* (2012) 9:511–528
- [28] Bonham-Carter, G., 1994. *Geographic Information Systems for Geoscientists: Modelling with GIS*. Pergamon press, Oxford UK/Burlington, MA.
- [29] Carranza, E.J.M., Hale, M., 2002a. Spatial association of mineral occurrences and curvilinear geological features. *Mathematical Geology* 34, 203–221.
- [30] C J Van Westen , 2009, Tools for map analysis applied to the selection of a waste disposal site, <ftp://ftp.itc.nl/pub/ilwis/pdf/appch18.pdf>
- [31] Wahyu Tjatur SA, Ira Prasetyaningrum, Tri harsono , Shiori Sasaki, Yasushi Kiyoki, 2013, Demam Berdarah dalam Perspektif Urban : Analisa Statistik untuk Awareness Strategy, "The National Conference on Smart-Green Technology in Electrical and Information System (CGTEIS)" 2013, Proceeding CGTEIS 2013 page 23-31
- [32] Misael Enrique Oviedo Pastrana, Rachel Lage Brito, Rafael Romero Nicolino, Camila Stefanie Fonseca de Oliveira, João Paulo Amaral Haddad, 2014, Spatial and statistical methodologies to determine the distribution of dengue in Brazilian municipalities and relate incidence with the Health Vulnerability Index, *Spatial and Spatio-temporal Epidemiology*, Available online 13 April 2014
- [33] Dirk Brockmann, Vincent David and Alejandro Morales Gallardo, 2009, Human Mobility and Spatial Disease Dynamics, [http:// www. Transportation . northwestern. edu docs/ research](http://www.Transportation.northwestern.edu/docs/research)

# Intelligent software support of the SCRUM process

Radoslav ŠTRBA<sup>1</sup>, Jakub ŠTOLFA<sup>2</sup>, Svätopluk ŠTOLFA<sup>3</sup>, Michal KOŠINÁR<sup>4</sup>  
<sup>1,2,3,4</sup>*Department of Computer Science*  
*VŠB – Technical University of Ostrava,*  
*Faculty of Electrical Engineering and Computer Science*  
*708 33, Ostrava – Poruba, Czech Republic*

**Abstract.** Company success in a highly competitive agile environment depends on the speed of right decisions making process. The quality of those decisions depends on availability of statistical reports, results of simulation and other information gathered from various supporting software systems. This calls for establishment method that allows combining developed intelligent software tools. This paper presents a method for support of the SCRUM software process. We proposed the method for modeling and simulating of the software process model, with the aid of a neural network. In particular, we aim at improving software process development using results of the simulation and effort estimation.

**Keywords.** Agile Methods, Software Process, SCRUM, Neural Networks, Effort Estimation

## Introduction

This paper provides an overview of work being constructed in field of intelligent support of agile software processes. Our work is focused on the neural network based effort estimations and simulations of the scrum processes in commercial companies. It identifies three important questions: “Why simulate and estimate”, “What is the scope of the estimation and simulation model” and “How to simulate and estimate” – suitable simulation and effort estimation techniques in the specific agile environment.

We developed a method and intelligent software tools as a part of our research on agile methods, support of software processes modeling, estimation, simulation and executing. These tools allow modeling of the software process using formal methods. They also allow simulation and effort estimation for development of commercial software products.

## 1. Agile methods

The Change is what the modern software development is all about. Implementing the change might not be easy but agile methods respond to these requirements. An agile viewpoint is that there should only be the bare minimum of documentation [2] and the production of the actual software as its primary goal. Most of agile methods advocate test-driven development, which states that tests must be written before the code itself.

In addition, every program feature must be covered with tests. The idea is to develop the simplest solution that could possibly work for the current feature. Agile methods help the quality checking practices happen with a high frequency. [1].

## **2. The Monte Carlo Simulation**

The term “Monte Carlo simulation” was first coined in the Manhattan Project during World War II, because of the similarity of statistical simulation to games of chance played in the Monte Carlo Casino. This illustrates that that already in the 1940s people was using computers to simulate processes (in this case to investigate the effects of nuclear explosions). The simulation is particularly attractive since it is versatile, imposes few constraints, and produces results that are relatively easy to interpret. Analytical techniques have other advantages but typically impose additional constraints and are not as easy to see. Therefore, it is no surprise that in the context of Software Process Management, simulation is one of the most established analysis techniques supported by variety of tools. [4]

## **3. Neural networks**

In the past, most effort- and cost-modeling techniques have relied on algorithmic methods. That is, researchers have examined data from past projects and generated equations from them that are used to predict effort and cost on future projects. However, machine learning could be used for assistance in producing good estimates. For example, simulations based on neural networks can represent a number of interconnected, interdependent units, so they are a promising tool for representing the various activities involved in producing a software product. In a neural network, each unit (called a neuron and represented by network node) represents an activity; each activity has inputs and outputs. Each unit of the network has associated software that performs an accounting of its inputs, computing a weighted sum; if the sum exceeds a threshold value, the unit produces an output. The output, in turn, becomes input to other related units in the network, until a final output value is produced by the network.

There are many ways for a neural network to produce its outputs. Some techniques involve looking back to what has happened at other nodes; these are called back-propagation techniques. They are similar to the method we used with activity graphs to look back and determine the slack on a path. Other techniques look forward, to anticipate what is about to happen.

Neural networks are developed by “training” them with data from past projects. Relevant data are supplied to the network, and the network uses forward and backward algorithms to “learn” by identifying patterns in the data. For example, historical data about past projects might contain information about developer experience and the amount of effort required to complete a project. More details to be found in [Vondrak\_NS]. [3]

#### 4. Method for intelligent support of the SCRUM software process

The Software process simulation is increasingly being used to address a variety of issues from the project management of the software development. The simulation is used to support of the process improvement and to the software project management training. In this paper, we will focus on the short time span simulation application. [5] One of the issues is estimate the effort of development of the software product.

##### 4.1. Reasons for a simulation and an effort estimation

There is a wide variety of reasons for undertaking simulations of software process models. In many cases, a simulation is an aid to decision making. It also helps in risk analyze and reduction and helps management at the strategic and operational levels. We have divided reasons for using simulations of software processes into three categories of purpose:

- Planning of the software project: The Project management planning can be supported by the simulation. It can help estimate the effort and cost of products. The Simulation can also help analyze a risk in initial planning phase. Project managers can use the simulation to predict a possible outcome if a proposed action.
- Improvement of the software process: In contrast to planning, here is an improvement of the process. A Simulation can be used to identify bottlenecks and result can be used to calibrate the model.
- Understanding of the software process: Using simulation tools with the output graphical interface we can visualize a process flow and help people to understand effect of changes in the process configuration. The simulation can helps people understand the inherent uncertainty in forecasting agile process outcomes and the likely variability in actual results seen.

In previous paragraphs we've provided examples of practical issues, which can be addressed with the simulation. When developing software process simulation models, identifying the purpose and the questions project management would like to address is important to defining the model scope and data that need to be collected. [5] , [19]

For the most software projects, a biggest component of cost is an effort. We must determine how many man-hours of the effort will be required to complete the project. The Effort is certainly the cost component with the greatest error ratio. We have seen how work style, project organization, ability, interest, experience, training and other employee characteristic can affect the time it takes to complete a task. Moreover, when the group of developers must communicate and consult with one another, the effort needed is increased by the time required for meetings, documentation and training. [5], [6]. We provide the intelligent method based on simulation using a neural network. It's very useful for the effort estimation of the software product.

#### 4.2. *The Scope of the SCRUM simulation model*

Proposed approach for the simulation and the effort estimation in SCRUM environment starts with an architecture. The architecture involves effort estimation components and learning rule used here is the Back-propagation algorithm. The effort estimation consists of the three inputs components which get inputs from the design document. The three components are: [6], [19]

- Actor components: This takes information about the actors involved in the system.
- Use-case component: This takes information about the Use-Case involved in the design document.
- TEF component: This takes the information regarding the technical and environmental factors involved in the system.

Very important is thinking of time span of the simulation process model. We can think of a short time span, (it means that software product is developing) less than 12 months and organization simulation breath of one scrum-team. If we want to answer the key questions, we need the result variables. Depends on the key question being asked, there can be many different variables devised as the results of a software process simulation. [5], [6], [19]

Result variables for software process simulation of SCRUM include the following:

- An effort (cost)
- An ideal duration of one iteration (planning of iterations)
- A schedule (developers-time management)
- A defect level (quality assurance)

Input Parameters:

- Manually set parameters (number of actors – developers, developers properties, amount of tasks – User Stories)
- Application log-data (historical information from project development)
- Generated values (utilize Monte Carlo Simulation approach)

#### 4.3. *The agile based approach for a simulation and an effort estimation*

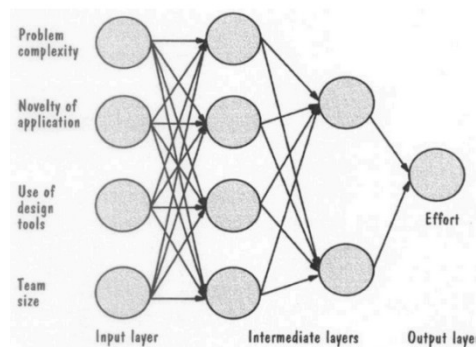
An agile environment is specific because there are some strictly defined parts of the process (defined using mathematical models, e.g. Petri Nets). On the hand there are some model-free parts of the process. For these model-free parts is an appropriate to use neural networks. However, the neural network procedure does not require a model specification, so an appropriate randomization of the network based synthesis of process realizations then allows for a Monte Carlo simulation.

For the simulation of strictly defined process models is appropriate to use a Discrete-Event Simulation (DES). [13]. The DES involves modeling a system as it progresses through time. A major strength of discrete event simulation is its ability to model random events (obstacles) and to predict the effects of the complex interactions between these events. Logical relationships link the different entities together and they are the key part of the strictly-defined simulation model, defining the overall behavior of the model. Entities are elements found in the real world and they may be either

temporary or permanent. Another key part of any simulation system is the simulation executive. The executive is responsible for controlling the time advance. A central clock is used to keep track of time. The executive will control the logical relationships between the entities and advance the clock to the new time. The simulation executive is central to providing the dynamic, time based behavior of the model.

The DES is efficient and particularly appealing when the process is viewed as a sequence of activities, such as in a manufacturing line where items or entities move from station to station and have processing done at each station. Discrete models easily represent queues and can delay processing at an activity if resources are not available. In addition, each entity has some properties. Changes to the attributes by the activities can provide much of the value of a discrete model [18].

A System behavior can change over time. For example, some key variables such as an availability of developers, or some customer requirements can change over time. In this case is appropriate to use the neural network based (model-free) simulation. A dynamic model is necessary when controlling these changes is important. Model of the dynamic simulation is very flexible and support modeling of a wide variety of system structures and dynamic interactions. The figure 2 illustrates how Shepperd [24] used a feed-forward neural network.



**Figure 1.** Example: Shepperd's feed-forward neural network

The designing of the network involves the selection of the network architecture. Proposed simulation for the scrum development phase implies the design of the network structure with two input nodes, two hidden nodes and one output node. The network gives effort in terms of the EMDs on the output layer which consists of only one node, the output node.

Training the network involves the compilation of the test data: the test data has been obtained by manually calculating the proposed simulation model. The training set is provided to the network designed as above. The acceptable error rate is set to 0.2. The network is trained with the compiled test data and the network converges over a period of thousands of iterations.

Proposed model accepts the number of variables and their occurrences, complexity of the code, criticalness of the code as the input and it computes the necessary values and provides it to the network. The network produces an output in terms of elementary mental discriminations (EMD). [19]

## 5. Model creation and configuration of the neural network

There are many modeling techniques for the process modeling, for instance Madachy (2008). However, the software process development has some specific features that must be taken into account (Kaufmann, 2006) and it has been characterized as “the most complex endeavor humankind has ever attempted” by Scacchi (1999). Nonetheless, a software process can and should be modeled in a formal way (Laurent, 1984). For the design of the simulation model of the SCRUM process, we’ve used semi-formal UML notation. [16], [17], [18]

### 5.1. Entities of the SCRUM process model

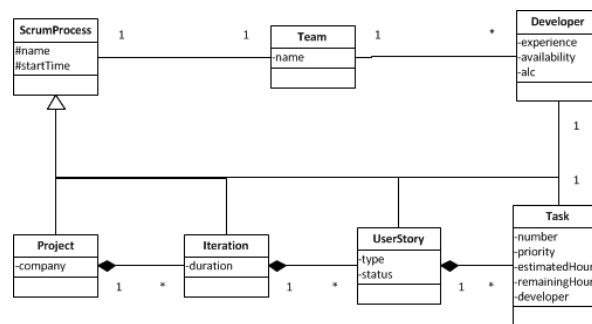


Figure 2. Class diagram of SCRUM process simulation model

We’ve created 4 basic types of entities that are fundamental for the SCRUM process models: Project, Iteration, User Story and Team involving developers. A Project contains a set of Iterations. Iteration contains a set of User Stories. The SCRUM follows an iterative approach to development, using time-boxed cycles. Each release of the system is implemented through a predefined number of time boxed Iterations. Iteration has a start time, duration and, again, contains a set of User Stories. The User Story is divided into particular tasks for developers. Developers are the main actors/roles of an agile software development process. They are responsible for develop the whole system. In the model, each developer is characterized by a set of attributes: experience, availability and average lines code.

### 5.2. Design and configuration of the feed-forward neural network

We need to set the training size. It is the number of input values and ideal values for the first column of our input matrix. The training size of our input matrix is 100. This value is based on average work-hours records of the one development team per month. Also, we need to set constant for maximum of errors. It is possible to train this network less than 1%, but it takes several days of the training. We need "perfectly trained" network with a very low prediction error, so we set this constant to 0.05. In short, now we have two double-values matrixes (1. input values, 2. ideal values). Our neural network is composed of input layer, two hidden layers and output. Number of neurons of the first hidden layer is 6 and number of neurons of second hidden layer is 4. Number of neurons of hidden layers has been determined based on rule-of-thumb methods. These

methods say that number of hidden neurons should be between the size of the input layer and the size of the output layer, and number of hidden neurons should be less than the twice the size of the input layer. [19] Designed feed-forward neural network layers:

- Input layer: 10 neurons
- First hidden layer: 6 neurons
- Second hidden layer: 4 neurons
- Output layer: 1 neuron

There are many training methods that can be used for feed forward neural networks. We've choose the back-propagation method. An agile environment also meets the requirements for a Monte Carlo, and is thus chosen as a basis herein.

## 6. Execution of the simulation

Simulation executions were performed using input parameters given in Table 1 and Table 2. Historical data from information system logs have given in Table 3. The first part of these parameters has been used for the simulation based on Discrete–Event approach.

**Table 1.** Manually set parameters

Parameter name	Value	Other information
Number of tasks	150	
Number of developers	4	Table 2.
Minimum estimated hours	4 hrs.	
Maximum estimated hours	15 hrs.	
Typical iteration duration	12 days	

**Table 2.** Manually set parameters

Id	Experience	ALC	Availability
1	2	250	150
2	4	400	220
3	4	350	250
4	5	450	250

Historical data from database (Table 3.) of the project management tool have been used as input parameters for the training of a feed-forward neural network.

**Table 3.** Application log-data from

Entity	Properties	Number of records
Task	type, priority, remaining-time	5428
User Story	difficulty, story-type, owner	1342
Developer	task-ALC, bugs, day-availability	1836

These values are based on real data taken from development of a ‘Clinical Processes’ module of healthcare information system that is being implemented for university hospitals and bigger clinics. The number of tasks and typical iteration duration are constant values. A function for generation time duration of every single task considers two input parameters: Minimum Estimated Hours and Maximum Estimated Hours. The



collection of tasks is automatically generated based on real log-data from information system.

One or more actors execute each activity of the process. The team whose activities are simulate consist of developers and those stands as actors within the simulation platform (Table 2). These actors are described by three main attributes: first property, experience means practice in years, second one is ALC (Average Lines of Code per day) and second property availability means monthly working time. Some rules can utilize more attributes like developers' certifications, specializations etc. Once the static structure of the system to be simulated is fed into simulation platform, we can execute the simulation itself based on dynamic rules. All approaches we've mentioned before are able to learn based on results of previous steps of the system simulation.

**Table 4.** Result values of the SCRUM process simulation

Output parameter name	Count	Number of developers / 4 weeks
Task-estimated hours	852	4
Developer-lines code	8726	4
Task-remaining hours	948	4

Results of the simulated process (Table 4.) flow combining DES and the neural network based approach, including experience, availability and ALC of team's developers. Important factor that influences the process flow is generation of obstacles, which are taken from real software process execution and could be e.g. blackout or internet connection outage. Obstacles are generated in periods following an exponential distribution. Duration of typical development events follows normal (Gaussian) distribution. We used the Monte Carlo Simulation approach to generate unpredictable events. Output parameters show the estimated and remaining time of tasks worked on and average lines code of whole team. More detailed results of the simulation can be read from the vector of double values and displayed using burn-down chart. It is enough information to estimate the effort for the software project developed under the SCRUM in duration of development less than one year.

## 7. Conclusion and future work

In this paper we've introduced an approach to simulate the SCRUM software process and estimate the software project effort, both with aid of neural networks. After the brief introduction we've described fundamentals of the Monte Carlo simulation approach, neural networks, software process modeling discipline, a creation and simulation methods. Based on our research we introduced the approach we have implemented within the development of healthcare information systems for university hospitals and bigger clinics. The realization team of this software product works under the SCRUM and with a good simulation model and machine learning described in a combination of Monte Carlo and neural networks we could design and build a methodology and support tool for a development process simulation. The simulation can helps to management with estimation of an effort needed for development of some product parts, which are easy to get. They can optimize the process or plan better.

So far the tests of simulation tools and formal model we've build prove that this approach works even it still has some insufficiencies and not all simulations are precise

enough compared to reality. However these deviations are usually caused by external factors like unpredicted obstacles or changes to the team.

Future works should lead to more complex and precise model of an agile software process that is executed within the company and improved the simulation tool that will include more attributes and will be able to work with greater set of external factors that can influence the implementation of software products. We also consider a creation of the interface on some of existing modeling tools or creating our own modeling tool embedded into the simulation application. The tool will be more user friendly.

## References

- [1] K. Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000. ISBN 0201616416.
- [2] J. Highsmith and A. Cockburn. *Agile software development: the business of innovation*. Computer, 34(9):120–127, 2001
- [3] [Vondrak\_NS], I.: *Neural networks (czech)*. VŠB-TUO, Czech Republic, Ostrava, revision 2009.
- [4] J.A. Buzacott. Commonalities in Reengineered Business Processes: Models and Issues. *Management Science*, 42(5):768–782, 1996.
- [5] Marc I. Kellner, Raymond J. Madachy, and David M. Raffo. Software process simulation modeling: Why? What? How? *The Journal of Systems and Software*, 46(2–3):91–105, April 1999.
- [6] Beer M, Spanos P.D. *Neural network based Monte Carlo simulation of random processes*, 2005 Millpress, Rotterdam, ISBN 90 5966 040 4
- [7] Humphrey WS (1995) *A Discipline for Software Engineering*. Addison-Wesley Professional.
- [8] Scacchi W, Mi P (1997) *Process Life Cycle Engineering: A Knowledge-Based Approach and Environment*. *Intelligent Systems in Accounting, Finance, and Management* 6:83--107-183--107.
- [9] Workflow Management Coalition: <http://www.wfmc.org/> WfMC Web, 1999
- [10] Smith JM, Smith DCP (1977) Database abstractions: aggregation and generalization. *ACM Trans Database Syst* 2 (2):105-133. doi:10.1145/320544.320546
- [11] Machado EP, Caetano Traina J, Araujo MRB (2000) Classification Abstraction: An Intrinsic Element in Database Systems. Paper presented at the Proceedings of the First International Conference on Advances in Information Systems,
- [12] Vergidis K, Tiwari A, Majeed B (2008) Business Process Analysis and Optimization: Beyond Reengineering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on 38 (1):69-82.
- [13] Raffo DM (1996) Modeling software processes quantitatively and assessing the impact of potential process changes on process performance. Ph.D. thesis, Carnegie Mellon University.
- [14] Brooks FP (1987) No Silver Bullet - Essence and Accidents of Software Engineering (reprinted form information processing 86, 1986). *Computer* 20 (4):10-19.
- [15] Curtis B, Kellner MI, Over J (1992) Process modeling. *Commun ACM* 35 (9):75-90.
- [16] Madachy RJ (2008) *Software Process Dynamics*. 2nd edn. Wiley-IEEE Press.
- [17] Scacchi W (1999) Experience with software process simulation and modeling. *Journal of Systems and Software* 46 (2-3):183-192.
- [18] George S. Fishman. *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, Berlin, 2001.
- [19] Heaton Jeff. *Introduction to neural networks*, ISBN: 1-60439-008-5
- [20] Garg PK, Scacchi W (1989) ISHYS: Designing an Intelligent Software Hypertext System. *IEEE Expert: Intelligent Systems and Their Applications* 4 (3):52-63.
- [21] Mi P, Scacchi W (1990) A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes. *IEEE Trans on Knowl and Data Eng* 2 (3):283-294. doi:10.1109/69.60792
- [22] Mi P, Scacchi W (1996) A meta-model for formulating knowledge-based models of software development. *Decis Support Syst* 17 (4):313-330. doi:[http://dx.doi.org/10.1016/0167-9236\(96\)00007-3](http://dx.doi.org/10.1016/0167-9236(96)00007-3)
- [23] Kosinar, M., Kohut, O., Frydrych, T.: Architecture of intelligent agents in MAS, *Proceedings of European-Japanese Conference 2008*
- [24] Shepherd AJ. "Second-order methods for neural networks: Fast and reliable training methods for multi-layer perceptrons," London: Springer, 1997

# Generic Workflows - A Utility to Govern Disastrous Situations

Marina TROPMANN-FRICK <sup>a,1</sup>, Bernhard THALHEIM <sup>a</sup>, Diethard LEBER <sup>b</sup>,  
Clemens LIEHR <sup>c</sup> and Gerald CZECH <sup>d,e</sup>

<sup>a</sup> *Christian-Albrechts-University Kiel, Department of Computer Science,  
Christian-Albrechts-Platz 4, D-24118 Kiel, Germany*

<sup>b</sup> *Geoexpert Research and Planning GmbH, Brunhildengasse 1, A-1150 Vienna, Austria*

<sup>c</sup> *riocom - Consulting Engineers for Water Management and Environmental  
Engineering, Rivergate, Handelskai 92, A-1200 Vienna, Austria*

<sup>d</sup> *Software Competence Center Hagenberg, Softwarepark 21, A-4232 Hagenberg,  
Austria*

<sup>e</sup> *OÖ Landesfeuerwehrverband, Petzoldstrasse 43, A-4017 Linz, Austria*

**Abstract.** Damage caused by natural hazards is increasing all over the world. All countries intensify their efforts to predict and prevent hazard events and to decrease disaster impact. Disaster management is one of the challenging, complex and critical application areas dealing with hyper dynamic situation changes, high velocity, voluminous data and organizational heterogeneity. Successful management of disaster response requires flexible and adaptable solution techniques including very accurate, fast and dynamic activity guidance for supporting of process coordination, decision making and information logistics in real-time.

Systems currently existing are mostly inapplicable for this challenging application area due to their almost exclusive support of structured pre-defined processes in a static environment. We propose to use generic workflows that satisfy these complex requirements and that provide support for organizing processes and information flow in disaster situations thereby providing decision support to crisis managers and "first responders". We illustrate our approach in a sample landslide scenario observed in the Austrian Alps.

**Keywords.** Generic workflows, disaster management, process-aware information system, workflow management systems, generic functions, mini stories, generic workflows, decision making, continuous situation awareness

## 1. Introduction

Disaster management represents an important, versatile and critical domain. The course of action in disaster situations is never the same. Each disaster event is a formidable challenge to contingency management and emergency teams and requires intensive communication, coordination and immediate response to the changing situations. Usually, there are contingency plans, hazard maps and a strict authority line to manage a disaster,

---

<sup>1</sup>Corresponding Author: Marina Tropmann-Frick, Christian-Albrechts-University Kiel, Department of Computer Science, Christian-Albrechts-Platz 4, D-24118 Kiel, Germany, email: mtr@is.informatik.uni-kiel.de

but the situation and the response organizations are constantly changing, requiring fast adaptation and high flexibility.

Gathering of information and coordination of information flow is one of the challenging tasks in emergency situations. In real world scenarios emergency teams have to deal more often with incomplete, limited in quantity and quality or even missing information. Many information sources are not available for all involved teams and organizations at the time they need that. For example, detailed geographical or hydrological maps, hazard maps, contingency plans or data from diverse sensors are scattered over many different departments and institutions. The EU-project Monitor II [16,6,19,17,26,18] detected that the quality of the contingency planning procedure strongly depends on appropriate process and scenario information, when dealing with natural hazards. Currently, crucial information needed for contingency planners is not contained in hazard and risk maps. The Monitor II project also highlighted necessities for linking hazard mapping and contingency planning. The coordination of the information flow becomes very complex and time-consuming process for systems that assist management in hazard situations [3].

Typically, disaster management depends on a large variety of parameters, behavior of actors involved, organizations handling the situation and services. Meteorological services, medical services, police and fire services, regional contingency and natural hazard management must be integrated. Finally, hazard management guide teams while dealing with hazardous materials or potential epidemics. In some cases it is necessary to involve also other actors with specific knowledge about context, location or jurisdiction, e.g. the municipality, representatives from industrial complexes, or local safety and security staff. The resulting heterogeneity of e.g. knowledge, experience and interests in teams may cause conflicts in communication, information exchange or decision-making [7,34]. The centralized coordination is certainly one of the key requirements for successful disaster response management [11].

Most actions during a disaster response are not predictable and cannot be planned completely beforehand. So the corresponding processes cannot be prespecified and handled in a standard way. We introduce generic workflows for coordination of disaster management processes. They allow accurate, fast and dynamic activity guidance and information coordination in complex situations.

Generic workflows are flexible and adaptable workflows belonging to the area of process-aware information systems (in particular workflow management systems). Process-aware information systems support and control business processes [25]. We distinguish between business processes that can be automated and those that cannot be easily automated. In both cases, the workflow model must entirely represent all activities, all parameters and the full control and data flow. This model becomes executable. In the second case, however, non-automatable activities are given by worklists for specific responsible actor groups.

Workflow management systems are used to organize and control business processes [9]. They are especially applicable for structured processes with sequential or parallel activities which require coordinated processing and involve several actors with different roles. Typical workflow management systems can mainly be used in a static environment with clearly defined organizational structures, completely given business processes and full control. Workflows for such business processes are completely predefined at process design time. Exceptions are part of the workflow. Deviations are often described as separate

workflows. They must however be known at modeling time [10,27]. EPK and BPMN support such business processes and their modeling [20,30].

Process-aware information systems are going to be used in applications which demand higher flexibility [8,12,13,21,15,33,22]. For example, M. Reichert and B. Weber [25] survey approaches to manage dynamics in process-aware information systems. We can distinguish between design-time and run-time flexibility. Variability, adaptation, evolution and looseness are the four main categories of flexibility.

Disaster management processes are in the looseness category. They are non-repeatable (every process instance is different), unpredictable (there is no knowledge existing about situation changes during an event) and emergent (the processes emerge during execution when more information becomes available). The situation specific parameters are unknown in the beginning and might change during process execution. Because of the huge number of parameters and possibilities of process development, dynamic approaches can easily become too complex and incomprehensible for dealing with.

Disaster management processes are highly dynamic and there is no possibility to pre-define every exception or variation. Therefore static approaches are quite ineligible. Although existing dynamic approaches can deal with a certain degree of flexibility, they may fail because of the huge number of parameters and case variations that must be considered.

Our approach is based on the idea of genericity. It allows us to construct an abstract generic workflow which can be adapted to dynamic changes during the refinement process at runtime. We present the structure and basic components of generic workflows and show how our approach satisfies requirements of disaster management in a real world disaster scenario from the Austrian Alps. Our approach extends the research in [31] and [32].

Our solution is going to be used in the EU-project INDYCO [14]. The main goal of this collaborative project is the development of an INtegrated DYnamic decision support system COmponent for disaster management systems.

## 2. Disaster Management

Disaster Management can be defined as the organization and management of resources and responsibilities for dealing with all humanitarian aspects of emergencies, in particular preparedness, response and recovery in order to lessen the impact of disasters [2]. In disaster situations, the major challenge for authorities is the protection of life (human and animal), property, and the vital life-supporting infrastructure necessary for disaster mitigation. Any delay or laxity in disaster relief could escalate the magnitude of distress for the victims. Advanced disaster management technology could provide a critical support system for disaster management authorities at times of disaster-related crises.

Effective disaster management depends on the informed participation of all stakeholders. The widespread and consistent availability of current and accurate data is fundamental for correct decision-making at any stage of disaster management [28].

Since disaster management includes various tasks and measures it is necessary to reduce its complexity. Taking into account the dynamic situation changes occurring during disastrous events it is necessary to reduce the application area for our approach concerning the following criteria. This enables us to model a general disaster scenario and to imple-

ment an application with a limited number of solutions for a limited variety of disaster situations.

1. Phases of disaster management:

- Prevention/Preparation
- Warning
- Response
- Recovery

Most disaster events contain all those phases. Prevention together with preparation is mostly the starting phase for disaster management. Activities within this phase are geared to reducing the negative consequences of disaster events and exposure to disasters. Directly before an event happens there is the phase of notification and warning. Information gathering, analysis, monitoring, forecasting and information exchange are the most important activities that start in the warning phase and end only after the disaster.

Response phase comprises the activities during and immediately after the current disaster event, such as saving of human life, protection of important constructions, supply of goods and services, as well as the protection of the environment. After that comes recovery phase, the last phase for the current disaster event. This is the phase of restoring the affected area and involve rebuilding of destroyed property, reemployment and the repair of infrastructure.

We concentrate in our work on the **response phase** of disaster management. Quick and efficient actions and decisions are very important at this point. In order to make right decisions team leaders need support in form of centralized provision of information about the disaster, about the affected areas and possible situation development. Then they can provide right information and instructions to team members and helpers in the affected areas.

2. Categories of disaster:

- Flood
- Landslide
- Storm
- Fire
- Volcanic eruption
- Tsunami
- etc.

There are certainly many other categories of hazard events. Our concepts are not limited to any specialized category. But for the evaluation of our approach we concentrate at first on the categories flood, landslide and storm. Below in the section 4 we describe a case study for a landslide disaster scenario.

3. Separation in Continuous Situation Awareness and Workflow components. In order to manage the difficulties we are dealing with in disaster situations we distinguish in our research between two interrelated parts of disaster management processes.

- Continuous Situation Awareness (in the following CSA) has a focus on identification and assessment of situations. The CSA supports the workflow compo-

ment with information provision, collecting relevant parameters and describing the current situation.

- Workflow Engine handles information from the CSA component and adapts workflow execution to the current situation and situation changes. Our approach of generic workflows allows handling of dynamic changes and adaptation flexible during runtime.

### 3. Generic Workflows

As we described above disaster management has many highly dynamic issues, where it is not possible to predefine every exception or variation. Our approach of generic workflows uses the advantages of genericity and addresses the required adaptation and flexibility and thus allows us to handle the highly dynamic characteristics of disaster management.

#### 3.1. Genericity

The notion of genericity is not new. According to [1], genericity can be described as a quality to be not specific, typifying, applied to or characteristic of all members of a genus, species, class or group.

In our everyday life we come almost permanently upon generic activities. When we speak to someone, we adapt our way of speaking according to the person we are speaking to. Certainly we would speak to a child not in the same way we speak to a superior. There are also many other situations and activities, where we face and use genericity.

In computer science genericity is also widely used. A good example is the usage of generic algorithms in context of generic programming. In [23] the authors describe generic algorithms as parameterized procedural schemata that are completely independent of the underlying data representation.

We understand genericity as a capability to describe a group or class of objects on a certain abstraction level. This allows higher adaptation and flexibility.

#### 3.2. Generic Functions

The concept of generic functions is the central part of the dissertation of A. Bienemann [4]. The prototypical implementation provided by Bienemann is based on government and binding (GB) approach that was introduced by Chomsky [5]. Chomsky proposed a universal theory of languages. Basic concepts of the theory are the atomic units of the syntax. The GB theory assumes that a universal grammar can be split into two parts: levels of representation and a system of constraints. It assumes four different levels of representation and a derivational model [24,29].

Consider functions  $F, F_1, \dots, F_n$  of a chosen function algebra (introduced in [4]). Generic functions are functions

$$\mathcal{F} = (Dom, \phi, F, \psi, Rng), \quad (1)$$

with free configuration parameters, with predicates  $\phi$  for the domain  $Dom$  and  $\psi$  for the range  $Rng$  of  $\mathcal{F}$ . A derived function is generated from  $F = \theta(F_1, \dots, F_n)$  based on expression and instantiation of the configuration parameters and on instantiation of the predicates;  $\theta$  is here an  $n$ -ary operator  $\theta \in op^{\mathcal{F}}$  (set of function manipulation operators [4]). In [32] we described the approach of generic functions more detailed.

### 3.3. Mini Stories

The generic functions defined above are basic elements for generic workflows. They represent the atomic activities within a generic workflow and are indecomposable. Yet, one generic function alone cannot form a semantically logical unit, which would make sense for itself.

Thus, we introduce *mini story* as a semantically logical unit that is sufficiently small to be used flexibly in different scenarios and sufficiently large to be semantically self-contained. Mini stories represent abstract collections of generic functions which can be dynamically composed at runtime based on parameter initialization. The functions of the composition may vary; they can be replaced by semantically similar functions [31]. A mini story can be defined as a quadruple

$$\mathcal{M} = (\mathcal{F}, \mathcal{T}, \mathcal{S}, P), \quad (2)$$

where  $\mathcal{F}$  is a set of generic functions as defined above.  $\mathcal{T}$  is a set of transitions within a mini story.  $\mathcal{S}$  is a set of parameters defining the current state of the mini story. And  $P$  is a priority function.

Transitions are given as tuples of the form  $T_{ij} = (F_i, F_j) \in \mathcal{T}$  and can be represented as edges of a directed graph with node set  $\mathcal{F}$ .

The priority function  $P$  is defined as follows:

$$P : (\mathcal{F}, \mathcal{S}) \rightarrow \mathbb{R}_{\geq 0}, \quad (3)$$

and assigns each function  $F_i \in \mathcal{F}$  depending on the current state  $S_j \in \mathcal{S}$  a priority. The function with the highest priority will be selected for the next execution step.

The instantiation of a mini story is performed at runtime depending on the current state (and influencing parameters) during the execution of the priority function.

### 3.4. Generic Workflows

A generic workflow is an abstract, configurable and adaptable workflow. In [32] we already described in detail the construction and steering mechanisms of generic workflows. Therefore for the sake of completeness we give only a short overview about the most important mechanisms and refer for more details to our previous work. The construction elements include:

- Simple control statements: sequential execution, parallel branch, exclusive selection of branch, synchronization and simple merge.
- Extended branch and synchronization operations: multi branch, multi merge, discriminator, n-out-of-m assembly, synchronized assembly.
- Structural statements: repetition, implicit termination.
- Data related operations: static operations, which are executed according conditions which can be checked during compile time, operations, which are executed according conditions which can be only checked during runtime, operations with defined runtime assertions, operations with conditions for synchronization.
- State based operations: the delayed selection, the related parallel execution, the milestone oriented operation.
- Termination statements: termination operation, case termination.



We also enhance our previous specification of generic workflows by applying the concept of mini stories. We define a generic workflow as a collection of semantically indecomposable mini stories. In addition to generic functions, which represent the atomic activities within generic workflows, mini stories can be seen as semantically atomic components of generic workflows. Relating to this we must consider the composition rules for mini stories, which form an important part of the specification. These rules describe the conditions for composition of mini stories within a generic workflow. For example there can be rules that require the execution of some specific mini story after another or even as a successor of another specific mini story. There can also be rules defined for the prohibition of mini story execution in a specific order.

Mini story composition is a complex issue, that can be characterized by the following three general aspects:

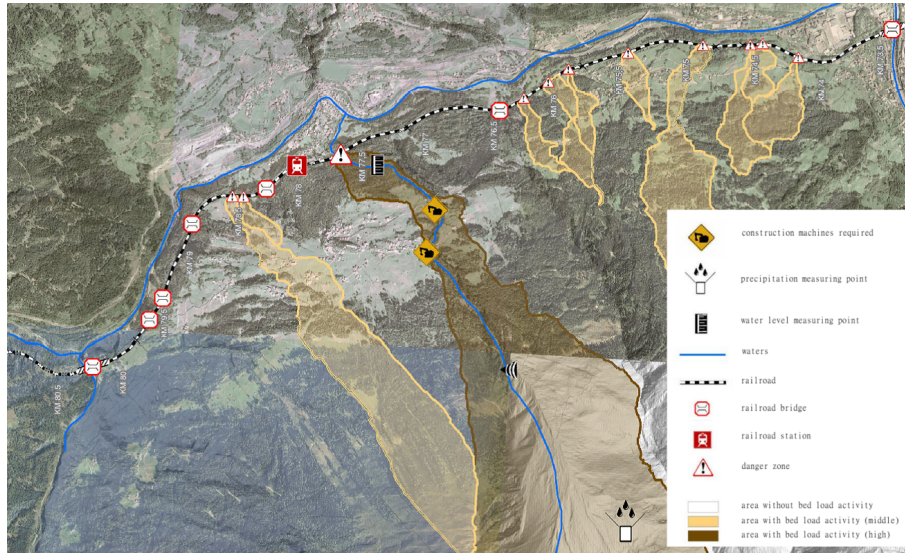
- The order of mini story execution is partly given by the execution order of generic functions. Depending on the priority function generic functions for the next execution step are selected. Therefore only those mini stories can be executed as next, which contain the selected generic functions and optimally start with one of them.
- On the other hand some rules for the composition are given by the context, where the execution takes place. The context of disaster management discussed in this paper possesses specific requirements and conditions. For the most disaster categories there are hazard maps, contingency plans or other guidelines existing (e.g. from the natural hazard management or municipality), which partly determine the order of mini story execution.
- The next important part are the influencing parameters. They can be characterized as configuration or control parameters. Some parameters belong only to one mini story and get their values allocated during the instantiation. Another parameters can be shared between various mini stories. As a consequence the instantiation of one mini story reduces the set of mini stories suitable for the next step and sets inevitably limitations for the instantiation of the following mini stories.

#### 4. Case Study

By this case study we want to illustrate how our ideas can be used in a practical way. In order to demonstrate our approach we selected one disaster scenario from a mountainous area in Austria.

*Case Scenario: Storm Rainfall Triggers Landslides in the Upstream Area of a Catchment - Consequently a Mudflow Hits Critical Infrastructure*

The basic scenario is characterized by heavy rainfall in a small mountainous catchment area. As a consequence landslides are triggered in the upstream area. Due to massive runoff the landslide material deposited in the torrent can be transported, a mudflow is developing. The mudflow impacts the railroad, destroying a bridge and derailing a train and blocking railroad traffic for quite some time. The **Figure 1** shows an example of an endangered area in the Austrian Alps.



**Figure 1.** Landslide danger area

The involved actors for this scenario are:

- natural hazard management (internal)
- regional contingency management team (internal)
- meteorological service provider (external)
- official hydrological service (external)
- federal service for avalanche and torrent control (external)
- first responders (external)

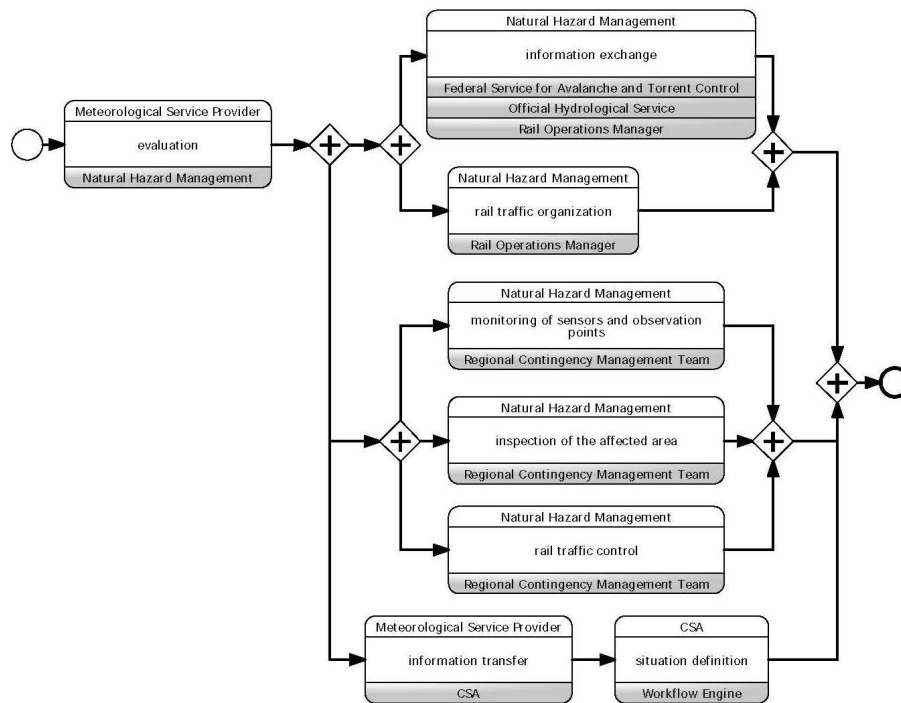
The following steps describe the situation development and actors interactions for the scenario:

1. The scenario starts with a storm front approaching the catchment area where the critical infrastructure is located. The responsible meteorological service provider generates weather forecast for this specific region. And because of threshold exceedance the meteorological service provider generates a warning and sends a message to pre-defined clients, in this case to the natural hazard management team of the railway undertaking, operating in the area possibly affected.
2. The next step for this case and for almost all disaster scenarios is information transfer to the involved teams and organizations. The natural hazard management team of the railway operator has to evaluate the situation. Depending on expected intensity and extent of the processes, they have to decide to involve other organisations/persons like e.g. the regional contingency management team members and rail operations manager.
3. From now on all involved teams monitor the situation and the weather conditions using web-based information platforms. Depending on the weather development the teams decide about the next steps.

4. In our scenario there is a weather deterioration, so it is expected that extreme precipitation will occur, possibly leading to a critical situation in the area observed. Therefore the regional contingency management team members meet and coordinate the information flow and the actions of other teams and organizations.
5. Due to the expected disturbance or disruption of regular operations the regional contingency management team/regional operations must prepare the redirection of trains and rail replacement services.
6. Next action is to gather information from the official hydrological service and federal service for avalanche and torrent control for the area possibly affected. At the same time existing hazard maps and contingency plans of the region are studied.
7. The next step is to send an expert team out on reconnaissance to the catchment area possibly affected. The information from this expert team in form of pictures, videos or text description will be analyzed and used for the next decisions.
8. Now the expert team in the affected area places observation points and mobile sensors there. The important part about it is to observe the railroad and to monitor the water level in the torrents. In parallel the regional contingency management team monitors the weather development.
9. To decide what to do next the regional contingency management team combines the external information with maps and hazard plans for this area and classifies the hazard scenario.
10. Based on the plans, weather forecast and other information the next action is to limit rail traffic to the minimum. Trains which have to pass the dangerous region should reduce their speed and inform the regional contingency management team about any problems.
11. In a next step the regional contingency management team receives information about increased precipitation in the area and after quite some time the gauging station installed in the torrent measures increasing water levels. Based on the pre-evaluation of possible hazard scenarios the regional contingency management team instructs the expert team in the field to inspect segments of the catchment with increased risk of landside activity. After some time the field team is reporting about a large landslide forming a landslide dam, blocking the drainage in the catchment. At the same time the gauging station is monitoring a fall of the water level.
12. In a next step of the scenario the landslide dam is breaking, and a mudflow is developed. The regional contingency management team sends alert information. Due to the short distance between the broken landslide dam and the railway infrastructure the warning time was too short. Therefore it was not possible to stop one train in time. The train is hit and badly damaged by the mudflow. The regional contingency management team sends reports to emergency teams and first responders( e.g. fire departments, red cross) and informs local and federal crisis management authorities.
13. Due to the scale of the disaster the local contingency management team decides to transfer the further crisis management operations to higher administrative levels. This action marks the end of the sample scenario. In a reality some static steps like a debriefing with each emergency team, a joint press conference and a

final debriefing of the contingency management team would end the emergency response phase.

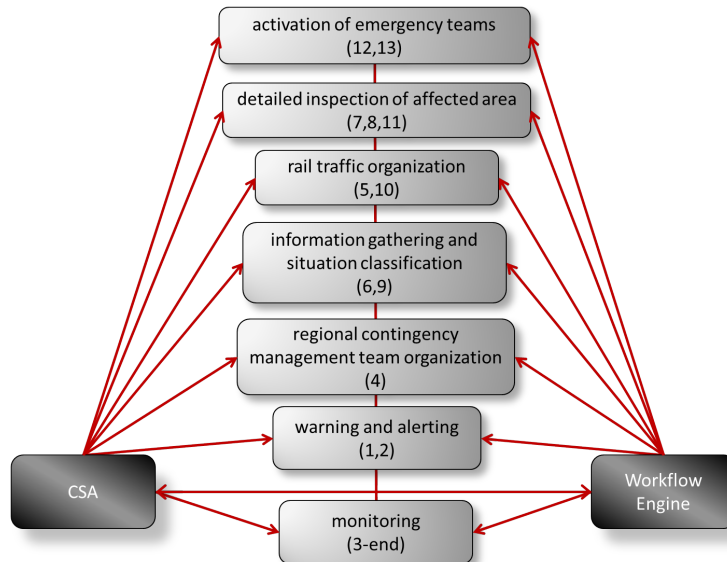
In order to handle the collaboration of the involved actors we modeled their actions and communication with a choreography diagram in BPMN [20]. The **Figure 2** gives an overview about the actors interaction for the scenario. Each box describes a communication part between the initiator (actor at the top of the box) and recipients (at the bottom of the box). The communication topic is located in the middle of each box. During the modeling process we consider also the separation in CSA and workflow component we made in section 2.



**Figure 2.** Actors interaction diagram for the landslide scenario

The next picture (**Figure 3**) shows a schematic representation of the workflow execution for this case study. The boxes in light grey represent possible mini stories for the given scenario. The numbers correlate to the enumeration in the scenario description. Dark grey boxes indicate the two disaster management process components, which coordinate the mini stories execution order. There is a permanent information exchange between CSA and Workflow Engine, thereby CSA component identifies situation changes and reports them to the Workflow Engine, which then reacts to the changes and activate the appropriate mini stories. The arrows demonstrate the execution flow of mini stories for the given scenario.

The workflow begins with the initial state described informally in step 1 of the landslide



**Figure 3.** Schematic mini stories representation for the landslide scenario

scenario. The CSA component identifies a dangerous weather situation in the described area and refer this information to the Workflow Engine. As a reaction the Workflow Engine triggers the mini story *warning and alerting*. The execution of this mini story forces the Workflow Engine to trigger the next mini story *monitoring*, which continues execution during the whole runtime of the workflow. The *monitoring* - mini story continuously gathers and analyzes information from different sources (sensor data, reports, etc.) and thereby changes the current state of the workflow influencing the mini story execution priority by the Workflow Engine.

In the similar way the workflow execution proceeds during the given scenario. Subsequent mini stories are triggered by the Workflow Engine. So for example the detection of the mudflow that hits a train (landslide scenario steps 12, 13) leads to the selection of a mini story with a highest priority for such case - *activation of emergency teams*. This mini story finishes the execution of the workflow for the described scenario. But it can also be seen as a starting point (starting mini story) for the next workflow, executed by the emergency teams. In this way the workflow can be continued or a new can be started with specific pre-defined parameters.

## 5. Conclusion and Future Work

Disaster management has many aspects that are difficult to handle with existing methods and standard processes. Many disaster scenarios are unpredictable and in their progress pass through various situation changes under time pressure and high risk. While existing approaches can successfully handle well-defined static processes, they may fail in more complex and dynamic situations.

The presented approach of generic workflows meets the requirements of the applications in disaster scenarios. In this work we introduce the structure of a generic workflow and

demonstrate our approach on a real world scenario. We plan to use the elaborated and here presented concepts as an essential part of a framework for the project INDYCO [14]. The goal of the project is to develop an integrated dynamic decision support system component (INDYCO) for disaster management systems, which will enhance existing disaster management and alarm systems. The project will provide users, contingency planners and civil protection services with support for decision making in complex and dynamically changing disaster scenarios.

### Acknowledgments

The authors would like to thank all partners from the INDYCO project for their support and contribution to the project.

### References

- [1] *Webster's Third New International Dictionary*. 1993.
- [2] IFRC - International Federation of Red Cross and Red Crescent Societies, <http://www.ifrc.org/en/what-we-do/disaster-management/about-disaster-management/>, accessed 22.01.2014.
- [3] N. Bharosa, J. Lee, M. Janssen, and H. R. Rao. A Case Study of Information Flows in Multi-agency Emergency Response Exercises. In *Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections Between Citizens, Data and Government*, dg.o '09, pages 277–282. Digital Government Society of North America, 2009.
- [4] A. Bienemann. *Context-Driven Generation of Specifications for Interactive Information Systems*. Dissertationen zu Datenbanken und Informationssystemen. AKA, 2008.
- [5] N. Chomsky. *Some Concepts and Consequences of the Theory of Government and Binding*. Linguistic Inquiry Monographs. MIT Press, 1982.
- [6] A. Corsini, S. Kollarits, D. Leber, J. Papez, T. Preseren, I. Schnetzer, and M. Stefani. MONITOR II - New Methods for Linking Hazard Mapping and Contingency Planning. <http://www.monitor2.org/>. 2010.
- [7] S. S. Dawes, A. M. Cresswell, and B. B. Cahan. Learning From Crisis: Lessons in Human and Information Infrastructure From the World Trade Center Response. *Social Science Computer Review*, 22(1):52–66, 2004.
- [8] C. Ellis, K. Keddara, and G. Rozenberg. Dynamic change within workflow systems. In *Proceedings of conference on Organizational computing systems*, COCS '95, pages 10–21, New York, NY, USA, 1995. ACM.
- [9] L. Fischer, editor. *Workflow Handbook 2003*. Future Strategies Inc., Published in association with the Workflow Management Coalition, 2003.
- [10] D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. In *Distributed and Parallel Databases*, pages 119–153, 1995.
- [11] R. González. *A Framework for ICT-supported Coordination in Crisis Response*. 2010.
- [12] Y. Han and A. Sheth. On Adaptive Workflow Modeling. In *Proceedings of the 4th International Conference on Information Systems Analysis and Synthesis*, pages 108–116, Orlando, Florida, July 1998.
- [13] P. Heintz, S. Horn, S. Jablonski, J. Neeb, K. Stein, and M. Teschke. A Comprehensive Approach to Flexibility in Workflow Management Systems. In *WACC99, Work Activities Coordination and Collaboration*, ACM Press, San Francisco, USA, february 1999.
- [14] INDYCO. <http://www.scch.at/en/service/news-events/9831>, 2012.
- [15] J. Klingemann. Controlled Flexibility in Workflow Management. In *Proceedings of the 12th International Conference on Advanced Information Systems Engineering*, CAiSE '00, pages 126–141, London, UK, UK, 2000. Springer-Verlag.
- [16] S. Kollarits, D. Leber, and I. Schnetzer. Linking Hazard Maps to Contingency Planning Needs - the MONITOR II Continuous Situation Awareness System (CSA). In *Geophysical Research Abstracts*, volume 13. EGU General Assembly EGU2011-8619-1, 2011.

- [17] S. Kollarits, D. Leber, and I. Schnetzer. Scenario Models as a Link Between Hazard Mapping and Contingency Planning. In *12th Congress INTERPRAEVERT*, 2012.
- [18] S. Kollarits, D. Leber, I. Schnetzer, A. Schwingshandl, and M. Ortner. New Approaches in Disaster Risk Management. In *International Symposium Urban Flood Risk Management (UFRIM), 21th-23rd September*, 2011.
- [19] D. Leber. The MONITOR II Approach. In *Proceedings of the PARAMOUNT Mid Term Conference, 07-08 June 2011*, pages 17–20, 2011.
- [20] L. Fischer, editor. *BPMN 2.0 Handbook Second Edition*. Future Strategies Inc., Published in collaboration with the Workflow Management Coalition (WfMC), 2012.
- [21] M. Momotko and K. Subieta. Dynamic change of Workflow Participant Assignment. In *ADBIS 2002. LNCS*. Springer, 2002.
- [22] R. Müller, U. Greiner, and E. Rahm. AgentWork: a Workflow System Supporting Rule-Based Workflow Adaptation. *Data and Knowledge Engineering*, 51(2):223 – 256, 2004.
- [23] D. R. Musser and A. A. Stepanov. Generic Programming. In *Lecture Notes in Computer Science 358*, pages 13–25. Springer Verlag, 1989.
- [24] N. Chomsky. *The minimalist program*. MIT Press, Cambridge, 1995.
- [25] M. Reichert and B. Weber. *Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies*. Springer, Berlin-Heidelberg, 2012.
- [26] F. Ronchetti, A. Corsini, S. Kollarits, D. Leber, J. Papez, K. Plunger, T. Preseren, I. Schnetzer, and M. Stefani. Improve Information Provision for Disaster Management: MONITOR II, EU Project. In C. Margottini, P. Canuti, and K. Sassa, editors, *Landslide Science and Practice*, pages 47–54. Springer Berlin Heidelberg, 2013.
- [27] N. Russell, A. H. Hofstede, D. Edmond, and W. van der Aalst. Workflow Data Patterns: Identification, Representation and Tool Support. In L. M. L. Delcambre, C. Kop, H. C. Mayr, J. Mylopoulos, and O. Pastor, editors, *24th International Conference on Conceptual Modeling*, pages 353–368, Klagenfurt, Austria, 2005. Springer.
- [28] S. Sahu. Guidebook on Technologies for Disaster Preparedness and Mitigation. Asian and Pacific Centre for Transfer of Technology (APCTT). 2009.
- [29] E. Stabler. Derivational Minimalism. In C. Retore, editor, *Logical aspects of computational linguistics*, volume LNCS 1328, pages 68–95. Springer, 1998.
- [30] D. M. Stephen A. White. *BPMN Modeling and Reference Guide*. Future Strategies Inc., 2008.
- [31] B. Thalheim and M. Tropmann-Frick. Mini Story Composition for Generic Workflows in Support of Disaster Management. In *Proceedings of the 24th international workshop on Database and Expert Systems Applications*, DEXA 2013, pages 36–40. IEEE Computer Society, 2013.
- [32] B. Thalheim, M. Tropmann-Frick, and T. Ziebermayr. Application of Generic Workflows for Disaster Management. In Y. Kiyoki, T. Tokuda, and N. Yoshida, editors, *Proceedings of the 23rd European-Japanese Conference on Information Modelling and Knowledge Bases*, Information Modeling and Knowledge Bases XXIII. IOS Press, 2013.
- [33] W. M. P. van der Aalst. How To Handle Dynamic Change and Capture Management Information? An Approach Based on Generic Workflow Models. *Comput. Syst. Sci. Eng.*, 16(5):295–318, 2001.
- [34] S. W. and D. J. A Case Study of Coordinative Decision-Making in Disaster Management. *Ergonomics*, 43(8):1153–1166, 2000.

# Mutual Resource Exchanging Model in Mobile Computing and its Application to Collective Intelligence 3D Movies

Naofumi YOSHIDA

*Faculty of Global Media Studies, Komazawa University*

**Abstract.** The only way for efficient use of limited resources is exchanging them each other in mobile environment. In this paper, an application for production of 3D movies by collective intelligence is presented. By this model, flexible and elastic usability can be implemented on mobile devices in mobile computing environment. And novel contents creation can be implemented by combination of this model, security, and collective intelligence.

**Keywords.** Mutual Resource Exchanging, 3D Movie, 3D Image, Collective Intelligence, Information Sharing, Security

## 1. Introduction

The resource exchanging is the key function because we have the limitation of the capacity for every single device in mobile computing environment. If there is no chance to generate resources, the only way for efficient use of limited resources is exchanging them each other inside a closed world, in mobile environment. Based on that background, a mutual resource exchanging model[1] in mobile computing environment are proposed. In this paper, an application for production of 3D movies by collective intelligence is presented. By this model, flexible and elastic usability can be implemented on mobile devices in mobile computing environment. And novel contents creation can be implemented by combination of this model, security, and collective intelligence.

Figure 1 shows the overview of a mutual resource exchanging model[1] in mobile computing. In the conceptual layer, all target devices are summarized as a group of devices inside a certain closed world in mobile computing environment. All resources are connected to the group of devices. In the physical layer, all devices are recognized and resources are owned by each device. If one device requires much resource than the amount of its own, resources owned by other devices are exchanged to the target device. Though these exchanging is performed for every kinds of resources, in the conceptual layer there is no exchanging and we have just one group of target devices.

## 2. Security Issues in Mutual Resource Exchanging Model

Security issues are required because we do not allow stealing resources from other mobile devices. We have many mobile applications such as movie playing[2],

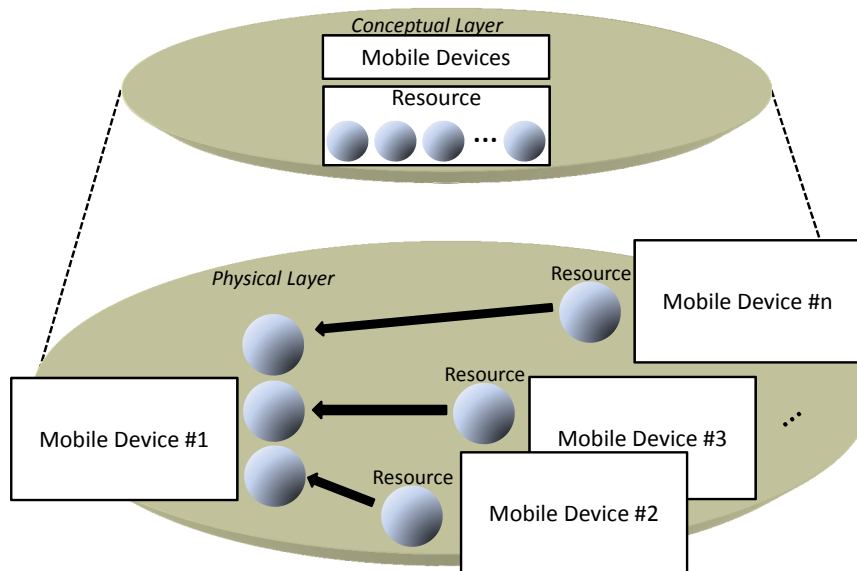


health care[3], context computing[5], cross-cultural communication[6]. And power management is still one of the major issues in mobile computing (e.g. [1][4][7]).

Table 1 shows the summary of the situation of security declaration from devices and the actions of results. The first four rows are simple situation. When a resource receiving device requests a resource, and a resource provision device permit provision, this resource exchanging is successfully activated. But when a resource provision device does not permit provision or declares that the resource is private, this resource exchanging is unsuccessful. Important points are mutual exchanging security declarations. From a resource provision device's point of view, it is natural to request some resource in return after the provision. When a resource receiving device requests a resource and permits a same kind of resource in return, and a resource provision device permits a provision of a resource and requests a same kind of resource in return after that, this mutual exchanging is successfully activated. But we have many kinds of resources, a resource receiving device will permit same or other or any resource to use in return because of the situation of resource usage. A resource provision device has also same situation. Table 1 shows the result action for the combination of a resource receiving device's situation and a resource provision device's situation.

### 3. 3D Movie Production by Collective Intelligence

When camera resources in mobile devices are mutually exchanged and combined with collective intelligence[13], novel contents creation can be implemented without specific devices. It's not dependent for specific devices, makers, or operating systems.



**Figure 1.** Overview of a Mutual Resource Exchanging Model in Mobile Computing [1]

**Table 1.** Summary of Security in Mutual Resource Exchanging:

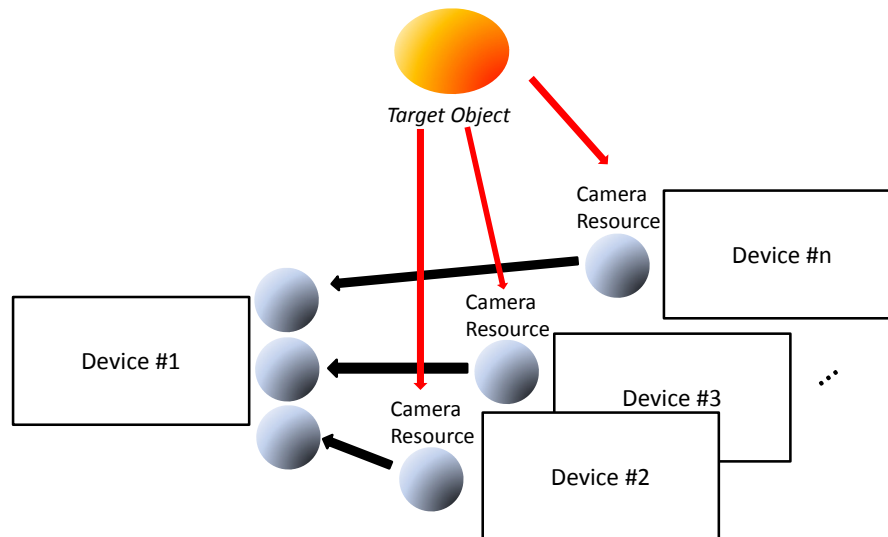
<b>Resource Receiving Device</b>	<b>Resource Provision Device</b>	<b>Action</b>
Request of Resource	Permission of Provision	Successful
Request of Resource	No Permission of Provision	Unsuccessful
Request of Resource	Public Resource	Successful
Request of Resource	Private Resource	Unsuccessful
Request of Resource and Permit Same Resource Provision when Ready	Permission of Provision and Request Same Resource in Return	Successful Mutually
Request of Resource and Permit Other Resource Provision when Ready	Permission of Provision and Request Same Resource in Return	Unsuccessful
Request of Resource and Permit Same Resource Provision when Ready	Permission of Provision and Request Other Resource in Return	Unsuccessful
Request of Resource and Permit Other Resource Provision when Ready	Permission of Provision and Request Other Resource in Return	Successful Mutually
Request of Resource and Permit Any Resource Provision when Ready	Permission of Provision and Request Same Resource in Return	Successful Mutually
Request of Resource and Permit Any Resource Provision when Ready	Permission of Provision and Request Other Resource in Return	Successful Mutually

Current 3D images or movies are produced by using a pair of fixed cameras. To provide people with 3D in a highly realistic manner, those two cameras need to be placed so as to align cameras' distance with the distance of people's eyes. It is practically difficult to produce 3D by using multiple different mobile devices like ordinary mobile phones or cameras as the position of mobile devices varies from time to time. The parallax value between the fixed cameras needs to be known. Furthermore, this technology requires special engines or stereo-cameras for producing 3D images or movies. Therefore it is difficult to popularize the conventional 3D and 3D streaming technology in terms of the cost. Thus, in this field we can't make 3D movies by using user's mobile camera as there is no proper real time collective intelligence based on live images and streaming movies being recorded separately.

We have related work on 3D graphics areas[14] and Peer-to-Peer computing areas[15].

This application provides a 3D image and movie producing method from still images or movie taken by several camera-equipped mobile phones as collective intelligence without specific fixed devices. By synthesizing two pictures/movies taken by different mobile phones located within a certain area (e.g. sport stadium, event site), this method enables to produce pseudo 3D pictures or movies and provide them to a third party remotely via the Internet. This application can be implemented on existing mobile phones. The mobile phones equipped with a camera, GPS receiver and 6-axis sensor take a picture of a certain target object, such as sport game, fireworks, auroras, and disasters like fires. These pictures are sent to a server with GPS data and 6-axis data. The server that received these pictures produces 3D images based on these pictures. These pictures have location data (e.g. GPS data) and direction data to the target object (6-axis data) so that the server picks up two images of which differences are the same as parallax. These images are taken by different cameras so that the mobile phone adjusts color, resolution, aspect ratio, frequency in images when 3D

images are played. The 3D images can be produced based on adjusted two images. The user can see the produced 3D movie data by a mobile phone, by using real time collective intelligence in mobile and ubiquitous environment without any specific devices for user.



**Figure 2.** Overview of this application: 3D Movie Production by Collective Intelligence

This application is new 3D image and movie producing system by mobile phones. Figure 2 shows an overview of the application. We assume to have a considerable number of people who have each mobile phone at a certain location, for example, in a sport stadium, event site, disaster site.

The mobile phone is equipped with a GPS unit, a 6-axis sensor and a camera. When a mobile phone user takes pictures or movies of a certain target (e.g. specific football player in a stadium, certain damaged building in a disaster site) by his phone, the pictures or movies are sent wirelessly and stored at the Internet (e.g. WWW server), and accordingly the address (URL) of the picture is stored with 6 axis data and GPS data at a managing server. The server classifies these pictures into each target object based on the GPS data. When the server receives a request from a third party who wishes to watch 3D images/videos remotely, the server selects two of the most relevant pictures or movies whose difference is the same as parallax by taking into account the positional information of the pictures/videos, produces pseudo 3D images or videos, and provide the requesting user with 3D. The target object should be taken by a considerable number of users, and further the object's pictures should be watched by substantial number of people. This system also has a point system to incentivize mobile phone users to take 3D images or videos.

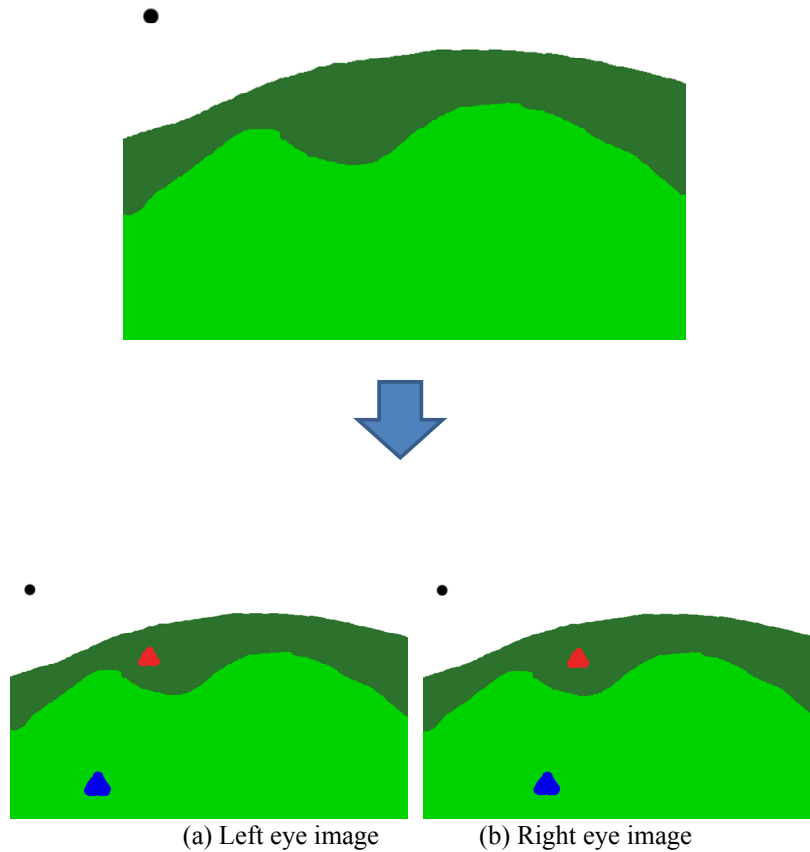
In the 3D movie providing side:

- 1) A lot of users are gathering together to see a target object (i.e. sport event, fireworks, etc). User takes pictures, or records a movie.
- 2) The information corresponding to the pictures and recorded movies is sent to the managing server.

- 3) The user providing pictures/movies earns points for every picture/live streaming he records. Also, more people see his picture/movie, more points he gets. He'll be given priority for frequency range when he watches 3-D images. As a consequence, the user of providing pictures would try to record better pictures/movie to earn more points.
- 4) The user will be able to use earned points for watching 3-D pictures/movies which are taken from opposite side.

In the 3-D movie receiving side:

- 1) A user can see a target object as 3-D image from a long distance by this system.
- 2) When the user requests 3-D image, he will specify target object, direction, place, or photographer.
- 3) The user can spend his earned point to view the 3D images.



**Figure 3.** Overview of this application: 3D Movie Production by Collective Intelligence

### 3.1. A mechanism for recognizing two images as a three-dimensional image due to parallax

Figure 3 shows a schematic diagram of 3D image production. For making a 3D image, two images are needed. One image is an image which can be seen an object from right eye, and the other image is an image which can be seen the same object from left eye. When a user watches these superimposed images, the user can see one 3D image.

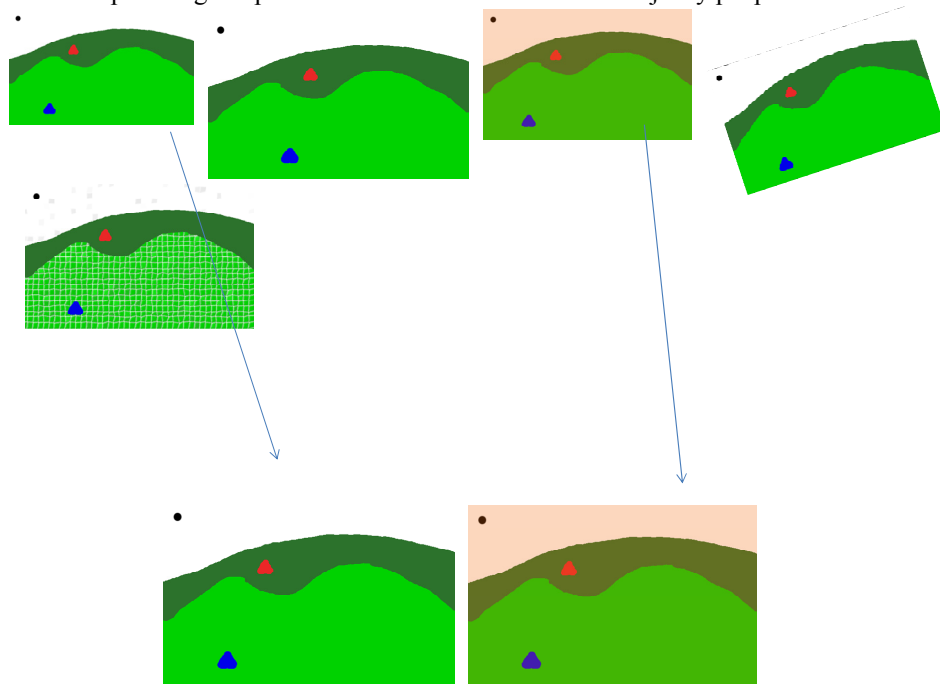
Namely, the difference of two images should be the same as parallax to make a 3D image.

### 3.2. A method for selecting two images which are recorded by different devices

The server selects two images by user indication.

- In case the user indicates a target objection, the server identifies images based on a tag of the image.
- In case the user indicates direction, the server identifies images based on a direction data.
- In case the user indicates a place, the server identifies images based on a GPS data.
- In case the user indicates a photographer name, the server identifies images.

Then, the server searches and chooses two images which difference is almost the same as parallax, as shown in Figure 4. It is said that everyone has different parallax value. This server keeps a range of parallax value which includes the majority people.

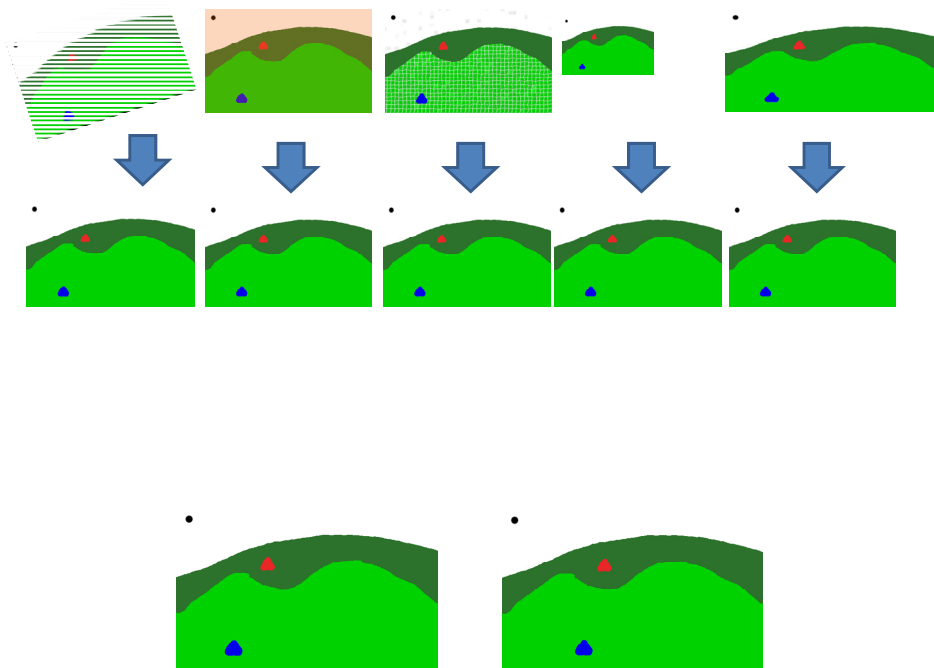


**Figure 4.** A schematic diagram of choosing two images for making a 3D image

### 3.3. A method for adjusting two images to enhancing a 3D image

Because each image is taken by different camera, it is necessary to make adjustments such as rotation, parallel shift of color, noise removal, zooming (resolution), trimming (aspect ratio). Figure 5 illustrates examples how pictures are adjusted. The upper part of the figure illustrates that the system conducts: 1) rotation, 2) parallel shift of color, 3) noise removal, 4) zooming (resolution) and 5) trimming (aspect ratio), for adjustment respectively, then acquires a pair of the two images illustrated at the bottom

of the figure. The mobile phone adjusts these differences based on the camera data, GPS, and 6-axis data which are received from the managing server.



**Figure 5.** A schematic diagram of adjusting two images

#### 3.4. Security Issues of this application

When all devices are permit all images and movies to use other devices and request same kind of resource after that in return, all devices can be 3D image and 3D movie recorder and receiver.

Advantage of this application is summarized as follows.

1. It can produce 3D image/movie by mobile phones from user's photos or movies without any special systems, using collective intelligence.
2. It can easily provide a 3D live streaming system by implementing a 3D synthesizer on the mobile phone.
3. It can collect and provide various images of the target objects from users by combining with a point system which promotes incentive to send images or movies.

#### 4. Conclusion

In this paper, an application of mutual exchanging model for production of 3D movies by collective intelligence is presented as its application. By this model, flexible and elastic usability can be implemented on mobile devices in mobile computing environment. And novel contents creation can be implemented by combination of this model, security, and collective intelligence. As future work, algebra formalization[12]

can be applied for this model. Also, qualitative and quantitative analysis of this model is important for the evaluation.

### Acknowledge

I would like to thank Prof. Yasushi Kiyoki for his valuable comments to this paper. I would also like to thank Dr. Shuichi Kurabayashi and Dr. Kosuke Takano for their valuable discussions about this paper.

### References

- [1] Naofumi Yoshida: A Mutual Resource Exchanging Model in Mobile Computing and its Applications to Universal Battery and Bandwidth Sharing, Proceedings of the 23rd European-Japanese Conference on Information Modelling and Knowledge Bases (EJC2013), pp.282-287, Nara, Japan, June 2013.
- [2] Pekka Sillberg, Shuichi Kurabayashi, Petri Rantanen, Naofumi Yoshida: A Model of Evaluation: Computational Performance and Usability Benchmarks on Video Stream Context Analysis, Proceedings of the 22th European-Japanese Conference on Information Modelling and Knowledge Bases (EJC2012), Prague, Czech Republic, June 2012.
- [3] Naofumi Yoshida, Daigo Matsubara, Naoki Ishibashi, Nobuo Saito, Norihide Ishikawa, Hikaru Takei, Shoichi Horiguchi: A Health Management Service by Cell Phones and its Usability Evaluation, Information Processing Society of Japan Transactions on Consumer Devices & Systems, Vol.2, No.1, pp.28-37, Mar. 2012.
- [4] Naofumi Yoshida, Naoki Ishibashi, Masaki Minami, Satoshi Washio, Daigo Matsubara, Nobuo Saito, Norihiro Ishikawa: A Hybrid Device Profile Detection Method and Its Application to Saving Electricity, Multimedia, Distributed, Cooperative, and Mobile Symposium 2012(DICOMO2012), 2012.
- [5] Anneli HEIMBURGER, Yasushi KIYOKI, Tommi KARKKAINEN, Ekaterina GILMAN, Kim KYOUNG-SOOK and Naofumi YOSHIDA: On Context Modelling in Systems and Applications Development, Information Modelling and Knowledge Bases, XXII, 17 pages, May 2011.
- [6] Anneli HEIMBURGER, Shiori SASAKI, Naofumi YOSHIDA, Teijo VENALAINEN, Petri, LINNA, Tatjana WELZER, Cross-Cultural Collaborative Systems: Towards Cultural Computing, Information Modelling and Knowledge Bases, Vol. XXI, pp.403-417, 2010.
- [7] Niu, L. and Quan, G., "System-Wide Dynamic Power Management for Portable Multimedia Devices". In Proceedings of the Eighth IEEE international Symposium on Multimedia (December 11 - 13, 2006). ISM. IEEE Computer Society, Washington, DC, 97-104.
- [8] Kevin W. Bowyer, Kyong Chang, Patrick Flynn, A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition, Journal of Computer Vision and Image Understanding, Volume 101 Issue 1, January 2006.
- [9] Douglas Lanman, Matthew Hirsch, Yunhee Kim, Ramesh Raskar, Content-adaptive parallax barriers: optimizing dual-layer 3D displays using low-rank light field factorization, ACM Transactions on Graphics (TOG), Volume 29 Issue 6, December 2010.
- [10] M. Forman, A. Aggoun, M. McCormick, A Novel Coding Scheme for Full Parallax 3D-TV Pictures, Proceeding of ICASSP '97, the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Volume 4, 1997.
- [11] T. Suenaga, Y. Tamai, Y. Kurita, Y. Matsumoto, T. Ogasawara, Image-Based 3D Display with Motion Parallax using Face Tracking, Proceedings of the 2008 IEEE Symposium on 3D User Interfaces (3DUI '08), 2008.
- [12] Naofumi Yoshida, Jun Miyazaki, "A Novel Approach to Time-Space-Direction Algebra for Collaborative Work in Ubiquitous Environment," In proceedings of International Conference on Collaboration Technologies (CollabTech 2006), pp.48 - 53, Jul. 2006.
- [13] James Surowiecki: "The Wisdom of Crowds: Why the Many are Smarter Than the Few and how Collective Wisdom Shapes Business, Economics, Societies, and Nations", Doubleday, 2004.
- [14] David Koller and Marc Levoy, Protecting 3D Graphics Content, Communications of the ACM, 48(6):74-80, June 2005
- [15] Rodrigo Rodrigues, Peter Druschel, Peer-to-Peer Systems, Communications of the ACM, Vol. 53 No. 10, Pages 72-82, 2010.

# Development And Usage Of A Process Model Corpus

Jürgen Walter, Tom Thaler, Peyman Ardalani, Peter Fettke, Peter Loos

*Institute for Information Systems (IWi) at the  
German Research Center for Artificial Intelligence (DFKI) and  
Saarland University  
66123 Saarbrücken, Germany  
firstname.lastname@iwi.dfki.de*

**Abstract.** In spite of the current research activities developing methods and techniques for business process model analysis, a standardized and digitally available process model corpus for evaluating these methods and techniques is still missing. Particularly with regard to a consistent appreciation of information systems, such a corpus is of high importance as it improves the development of standardized evaluations. Against that background, the paper at hand focusses the development of a process model corpus, whereby reusability is an important aspect. The corpus may serve as a standardized data basis for different application and analysis scenarios, e.g. for replicating prior research results or the evaluation of algorithms, methods and techniques and its further development. In order to realize these objectives the authors propose a procedure model which states as the basis for the developed process model corpus. This corpus contains reference models, models from practice and models from controlled environments and, in total, comprises 16 model collections with 2290 process models.

**Keywords.** Process models, corpus, metrics, reference model

## 1. Introduction

Nowadays, companies use large model databases to manage their business process models, which serve as a knowledge base for the design of their information systems. Oftentimes these databases contain several hundreds or even thousands of models [1] [2], wherefore methods and techniques for complexity reduction, handling and analysis of these data are needed. This demand is explicitly addressed by information systems research, e.g. in terms of process model similarity [3] [1], identification of structural analogies [4] [5] [6] or inductive reference modeling [7]. At the same time, an access to real process models from practice is missing, which is often caused by legal aspects or privacy. Companies are afraid of losing their competitive advantage by publishing their business processes.

Indeed, there are several approaches focusing on the conceptualization and the establishment of open access model repositories [8] (apromore.org, openmodels.org, openmodels.at, prozoom.ch), but concrete digital and processable models are very rare. Some trends in that direction can already be observed within the information systems research, e.g. in terms of the interest of the Business Process Management Conference



(BPM) in publishing the source code of software tools and implemented algorithms named in the proceedings. In that context, the possibilities of replicating the published findings are of major interest. Nevertheless, publishing the underlying data material is rarely focused. Though, particularly these data are essential for the replication and therefore of high importance for the research progress. The capabilities of corresponding corpora can be observed in different fields of research. E.g. the use of speech and text corpora in the fields of computational linguistics [9] [10] led to high benefits in speech processing, human computer interaction and automatic translation techniques. The use of genomic databases caused substantial progresses in the fields of biology, chemistry and medicine.

The paper at hand makes a first step towards a process model corpus which contains process models in a standardized, digital and processable format. Therefore, the authors developed a procedure model which serves as the basis for a prototypical implementation of the corpus. The aspects of digitalization and capturing as well as harmonizing process models are of high importance.

After this introduction, section 2 presents the vision and the context of the paper at hand as well as some application scenarios. Section 3 describes a general procedure model for the creation of model corpora. The instantiation of that procedure model for the creation of the focused process model corpus is then described in section 4, while section 5 introduces the developed corpus. The paper closes with a discussion and an outlook on the next steps in section 6.

## **2. Long-Term Research Objectives – A Vision**

The authors' vision is to develop a comprehensive model corpus which contains models in a standardized, digital and processable format. Thus, the following research objectives are focused: (1) Creating a consistent understanding of business application systems in different domains, (2) reusing the contained models in other contexts, (3) creating a homogeneous data basis for different application and analysis scenarios. The corpus should also be published for a free use in science. However, this highly depends on the license holder of the corpus content. Finally, the authors aim at publishing the corpus in terms of open models; similar to the open source idea which was established in the context of software development during the last years.

The initial starting point for that intention is the currently existing reference model catalogue [11] ([rmk.iwi.uni-sb.de/](http://rmk.iwi.uni-sb.de/)). It contains 98 reference model entries with lexical data and meta data like the number of containing single models. However, this catalogue does not contain digitally processable models (in terms of the used modeling language or a consistent exchange format) and there are also no entries of individual models from different domains.

Apart from the practical aspects mentioned above, like the replication of research findings or the evaluation of methods, techniques and algorithms, theoretical questions can be addressed as well. Some examples are the creation of a consistent understanding of terms over different domains and the automatic identification of modeling rules and conventions while modeling. This may improve the further development of current modeling theories.

In order to present the range of applications and analysis, in the following, the authors introduce some concrete scenarios. This overview is neither concluding nor

comprehensive, but it should illustrate the benefit of the developed corpus for the information systems research.

- **Process Matching** describes the mapping of nodes of a process model to the nodes of another process model [12]. Corresponding approaches are used in the context of model search, process model similarity, reusability of model fragments or inductive reference modeling. By using a process model corpus, the following questions could be answered: (1) To which extent are automatic approaches able to find matches which are determined manually. (2) Are there elements or model fragments which are available in several reference models?
- Analyzing **structural analogies** focusses on the identification of similar or analogue structures within one or more models [4] [5] [6]. The following questions could be addressed: (1) Which structures can be observed frequently, which ones seldom? (2) Which structures match each other? Do specific structure sequences exist? (3) Are there different structures in different domains? (4) Is it possible to define content-independent process templates?
- A further scenario is the search of **process variants**, as it is likely that specific models occur in different reference models, e.g. models related to acquisition and distribution or models in the context of accounting. The automatic identification of such fragments or models would contribute to the development of a comprehensive reference model over different domains. Apart from that, the corpus offers the possibility of reproducing the evolution of models, as model versions of different years can be analyzed.

### 3. Procedure Model for Corpus Development

#### 3.1. Introduction

In the context of the paper at hand, a corpus is defined as a structured and versioned library of models and model collections. Model collections, e.g. the SAP-R/3 reference model, cover several single models (in case of the named SAP-R/3 reference model: 604).

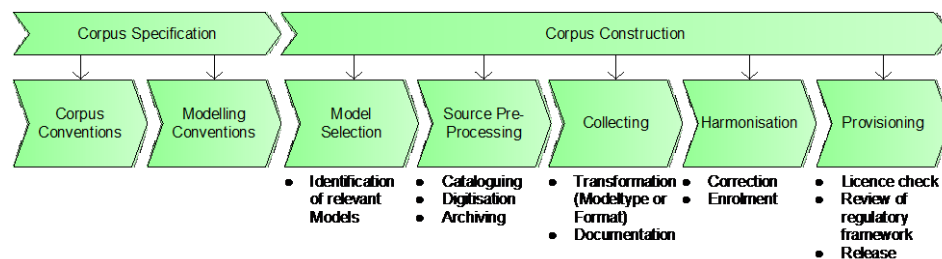


Figure 1: Procedure model for corpus development

In order to develop a model corpus, the authors developed a procedure model (Figure 1), which is described in the following. It covers the specifications of the corpus and the procedure which is needed for setting the content. The setting includes the model selection, the preprocessing of different sources as well as the process of gathering,

harmonizing and providing the models. The following principles should be considered during corpus development:

- **Principle of openness:** The corpus should be specified and the specification should be publicly available. This does also apply to the corpus' content.
- **Principle of powerfulness:** The corpus should be flexible regarding the supported modeling languages and it should completely implement the defined requirements.
- **Principle of stability:** The corpus should be stable in terms of further development. Changes of specifications should not lead to incompatibilities between different versions of the corpus.
- **Principle of tool support:** The corpus and its content should be processable by established software tools in the focused domain.

### 3.2. Corpus Specification

A concrete implementation of a corpus is controlled by the definition of corpus conventions and modeling conventions. The corpus conventions cover all rules and general requirements, which affect the whole corpus. The modeling conventions specify all relevant model aspects (see Table 1). Models are distinguished between source models, as-is models and to-be models. Source models are the models of the source document. Necessary transformations of the source model needed to fulfill the minimum requirements (related to the conventions) lead to the as-is model. The to-be-model results from the as-is-model after further transformations in the context of harmonization.

The treatment of concrete problem statements is one of the challenges in the application of analysis algorithms on models. E.g. some algorithms are developed for very specific conventions, which leads to the effect of an inadequate usefulness in other contexts or models. Thus, one should ask whether models should be included in the corpus as they are in source, or whether harmonization steps are necessary. The modeling conventions define how and which model constructs are being recorded or how they should be transformed. They contain not only the specifics of the chosen modeling language but also the handling of content and layout. These commitments ensure the consistent recording and handling of periodic modeling constructs, independent from the modeler or the model source. Otherwise, artificial differences could be constructed. These are hard to reconstruct, sometimes this might even be impossible. Thus, the modeling conventions depend on the corpus conventions.

In contrast to the *specification of modeling language*, it could be meaningful to globally define content and layout aspects in the corpus conventions. E.g. it could be the case that corresponding models of different model types are being focused (for example data and process models). Then, it would not be meaningful to harmonize the technical terms for the models of a model type.

Table 1: Corpus and modeling conventions

conventions of corpus	<b>Model types</b>	Which types of models should be accepted by the corpus? E.g. data models, process models.
	<b>Modeling language</b>	To ensure a consistent data basis, a modeling language for each model type must be determined. This is obligatory for all models of the same type, e.g. EPC, BPMN or Petri Net in case of process models.
	<b>Transformation between modelling languages</b>	Transformation rules or tools with which models of other modeling languages can be transformed into the chosen target language must be defined. As there are several languages for all types of models, an ad-hoc definition of transformation rules at the first appearance seems to be meaningful.
	<b>Exchange format</b>	An adequate exchange format (e.g. AML, EPML, PNML) must be chosen. This format should then be used to provide all models.
	<b>Other models</b>	It should be clarified how models of not focused types are to be handled. This is especially important for models which are already available in a digital and processable format.
	<b>Kind of appropriation</b>	It should be defined, in which way the corpus should be provided. Some possibilities are web publishing, version management systems like SVN or Git, single files or file packages like ZIP or RAR.
Conventions of modeling	<b>Specification of modeling language</b>	
	<b>Syntax</b>	Considering different models, it can be observed that the used constructs can diverge. Thus, all relevant model constructs which cover the essential modeling language of the corpus should be documented. It should also be defined if and how to correct syntax errors.
	<b>Execution semantics</b>	It should be defined, if and how to correct execution semantic errors like deadlocks or livelocks.
	<b>Transforming other constructs</b>	Transformation rules, which are used to transform (syntactically or semantically) unsupported constructs into supported constructs, should be defined. E.g. how to transform the SEQ operators to regular EPC constructs.
	<b>Content specification</b>	
	<b>Operational semantics</b>	It should be defined if and how operational semantic errors are to be corrected.
	- <b>Technical terms</b>	It should be determined whether technical terms have to be harmonized across all models in the corpus or if they have to remain domain-specific and model-specific.
	- <b>Acronyms</b>	It should be determined whether acronyms have to be resolved or if they have to remain unchanged since they often depend on the context or the domain.
	- <b>National language</b>	It should be determined whether the used national language of the models, e.g. English or German, remains unchanged or whether it must be translated into a consistent and/or a further national language.
	- <b>Spelling, punctuation and grammar</b>	It should be determined whether mistakes in spelling, punctuation and grammar have to be corrected or not.
	<b>Layout specification</b>	
<b>Layout</b>	It should be determined whether the layout of the source has to be ignored, taken or harmonized according to the specific requirements.	

### 3.3. Corpus Construction

**Model Selection.** Sources are documents of each type, which contain models or model collections of a defined type that is specified in the corpus conventions. For instance, if the model type is specified as process model then only process models are added to the corpus. Further selection criteria can be defined. The model selection itself as well as the defined criteria must be documented in the corpus conventions.

**Source Pre-Processing.** In a first step, it must be checked whether a source is published. If so, the source has to be catalogued. Basically, sources can be distinguished between analogue and digital sources. Analogue sources are printed documents like books, journals or conference proceedings that have to be digitized (mainly by scanning). A digital source is a document in a machine readable format. Source formats can be distinguished into non-processable formats (e.g. PDF) and directly processable formats (e.g. AML-, EPML or PNML-Model files). Analogue sources, as well as non-processable digital formats, have to be digitized or transformed into processable formats. The processable digital formats can be inserted into the corpus directly. For a better traceability it is recommended to archive the source. Implicitly applied archiving rules should be explicated and documented in the corpus conventions.

**Collecting.** In the collection phase, the selected models have to be processed in accordance with the minimum requirements defined in the model selection phase. Since the source models may be in different modeling languages and for each model type a unified modeling language is defined in the corpus conventions, the source models have to be transformed into the selected target modeling language, if necessary. Along with the data storage in a uniform exchange format, format transformations may also be necessary. In principle, transformations can be carried out both manually and automatically with tool support. A combination of both is also possible. The results of the collection phase are as-is models. If a source model already complies with the minimum requirements, the as-is model is equal to the source model. The as-is model can now be added into the corpus.

It should always be considered that all model transformations may have side effects. For example, information about links between models can be lost, when element names are changed unilaterally. A reconstruction of such induced faulty links can, if it is possible at all, only be realized in the aftermath and with much effort. This case occurs in particular when certain types of models (e.g. data model) or individual model components (e.g. ERM elements in EPCs) have been excluded; however, they have to be added later. Therefore, an adequate documentation of all transformations is recommended.

**Harmonization.** In the phase of harmonization, the as-is models, which are already available in a unified modelling language and in a single target format, have to be transformed into to-be models if they violate the modelling conventions. If no violations occur, the as-is model is equal to the to-be model. The new to-be-model has to be added to the corpus. In this phase, as in all other phases, the transformations have to be documented.

**Provisioning.** The legal framework has to be considered before any publication of the models. This typically affects the licencing law as well as the copyright of the authors of the sources.

## 4. Corpus Development

### 4.1. Introduction

In order to give the presented vision a form, the authors use the method of vertical prototyping. In this first iteration, only process models are focused. The developed process model was utilized to meet the objective. This leads to a narrow, but also diverse, corpus, which ensures that all included models are conform to the defined conventions.

### 4.2. Corpus Conventions

The Event-driven process chain was defined as the central modelling language and the process models as the model type. By the occurrence of the first cases of transformation, rules for other modelling languages were formulated and documented. The ARIS Toolset 7.2 was used for manual modelling and the ProM 5.2 tool when possible for transformation purposes. However, ARIS does not allow modelling EPCs, in which an event is immediately followed by another event. Such constructs considered as syntactically incorrect for EPCs, are even present in well-established reference models such as the SAP-System R/3. 'Dummy' functions were added in between the events in order to correct the EPCs. Nevertheless, with regard to the diversity of supported constructs, the ARIS Toolset is more appropriate. It offers the possibility to define one's own constructs, for example as needed for modelling SEQ connectors. As exchange format, the ARIS XML export format (AML) was chosen. Even if this exchange format is not openly specified, it is widely used in the academic area and by many software tools such as ProM. The stability of the format through various ARIS versions is not assured, thus, it can be seen as a disadvantage. Unlike other tools or modelling language-specific formats, such as PNML or EPML, it provides expansion capabilities through the definition of own construct. Moreover, it supports many different model types. Transformation rules for other exchange formats were also formulated and documented. Even model types such as value chains in the ITIL reference model (which were not relevant for our study), were added to the corpus, but not further considered. As a part of a prototype, the corpus is provided as a ZIP archive with the corresponding version information in each version.

**Modeling Conventions.** The different types of nodes (events, functions, hierarchized functions, process interfaces, XOR, OR and AND connectors) and directed edges were defined as elements carrying out (supporting) the information, both representing the control flow. Additionally, the item identifier, in which the substantive aspect of the models is encoded, was recorded. Rules for the transformation of constructs which were not supported were formulated and recorded by their first occurrence. Nevertheless, errors in the execution semantics (e.g. livelocks or deadlocks), as well as technical semantic errors, were not adjusted.

Some information, like shapes, colours, layout, and other graphical elements used, e.g. for the representation of organizational units or data elements, textual annotations and explanations, was excluded. . In order to avoid some adjustment, the language of the source was chosen as national language. While adjustments regarding the punctuation and the grammar were made, technical terms and acronyms remain unchanged.

### 4.3. Corpus Construction

**Model Selection.** As defined in the corpus conventions, only process models were used for further processing.

**Source Pre-Processing.** The models added to the model corpus were derived from various sources, such as books, so that most of the reference models were reproduced. Small recordable models from journals or conference proceedings can serve for the evaluation. Such published model sources are first classified. Another source of models resulted from individual process surveys. Usually, transcripts or audio recordings were at the modeler's disposal.

**Collecting.** The analogue or non-processable digital sources were recorded with the ARIS Toolset 7.2 as defined in the corpus convention and ProM 5.2 was used to transform digitally processable sources which were not conform to the exchange format.

**Harmonization.** Since there are no software tools allowing automatic adaptation of models with respect to the current conventions so far, this step was performed manually. The corpus-specific modeling conventions described above served as the basis for that.

**Provisioning.** The consideration of legal aspects was not a part of the work on the prototype.

## 5. Resulting Model Corpus

### 5.1. Corpus Scale

Based on the origin and type, each model collection or each model within the developed model corpus can be allocated to exactly one of the following three categories:

- **Reference models:** Reference models generally consist of descriptive and prescriptive model elements [FB08]: In a descriptive sense, a reference model captures similarities of a category of companies. In a prescriptive sense, a reference model presents a proposal for the design of enterprises.
- **Individual models:** Individual models describe processes in specific organizations. These include business models as well as models in public administration.
- **Models from controlled modeling scenarios:** A situation based on a textual description will be modelled from different test persons. This textual description helps the test persons to have both a common understanding of the problem and a uniform terminology. Therefore, resulting models are controlled models.

Table 2: Overview: developed process model corpus

C	Name   S   A   F	V	Remarks	L	#
R	ECO-Integral   [13]   a   book	I	Information systems for environmental management. Contains 38 EPCs and 11 function trees (as EPC).	de	49
		S	Contains EPCs only. Intermediate process interfaces are transformed into hierarchical functions. 3 EPCs are composed into one EPC for syntactical reasons.	de	36
R	Retail-H 1996   [14]   a   book	I <sub>1</sub>	Handelsinformationssysteme. Edition 1996. Contains 54 EPCs and 2 event hierarchies (as EPC).	de	56
		I <sub>2</sub>	Correspondent to the I <sub>1</sub> variant with transformed SEQ operators.	de	54
R	Retail-H 2004   [15]   a   book	I <sub>1</sub>	Handelsinformationssysteme. Edition 2004. Contains 58 EPCs and 2 event hierarchies (as EPC).	de	60
		I <sub>2</sub>	Corresponds to I <sub>1</sub> -variant, but with transformed SEQ-Operator.	de	58
		S <sub>1</sub>	Based on I <sub>1</sub> -variant with integrated event hierarchies and further structural adaptations.	de	58
		S <sub>2</sub>	Based on I <sub>2</sub> -variante with integrated event hierarchies and further structural adaptations.	de	58
R	ITIL   Bought from Software AG   d   ARIS-DB	I	Reference model for the IT Service Management. Digitisation is based on [16-20] by provider. Contains 19 EPCs, with an example for explanation, and further 297 models of other types.	de en	19
R	SAP R/3 1998   [21]   a   book	S	SAP R/3 reference model. Literal, syntactical and referencing errors corrected.	de	56
R	SAP R/3   source unknown   d   EPML	I	SAP R/3 reference model with cryptic model names and without hierarchies.	en	604
		S	Added plain model names and hierarchies.	en	604
R	Y-CIM 2.1   ARIS-Toolset   d   PDF	I	Reference model for industrial business processes. Contains the complete business model of the ARIS-Toolset 2.1a 1994 with syntactical corrections.	de	7
R	Y-CIM 1998   [22]   a   book	I	Reference model for industrial business processes. Covers EPCs and function trees; inclusive exercise EPCs and descriptions.	de	55
		S	According to I variant but without exercise EPCs and descriptions.	de	45
R	Y-CIM 1994   [23]   a   book	I	Structural correspondent to the German Y-CIM 1998. Labels and model names come from [23].	en	55
		S	Adaptions according to the German S variant.	en	45
I	Custom B2B   s   a   Text	I	Processes describing software customizing and the production of special machinery.	de	46
I	Business registration   s   a / d   text and audio	I	Business registration processes of 8 German communes.	de	24
I	GK-Rewe   [24]   d   PDF	I	Basic course "accounting" at Chemnitz University.	de	34
		S	Syntactical errors corrected.	de	34
I	E-Payment   s   a   Text	I	Electronic payment process of governance.	de	38
I	PMC   [12]   d   PNML	I	Birth registration processes of 9 countries and University admission processes of 9 German Universities. Originally modeled as Petri-Nets. PNML files were transformed to EPCs with ProM.	en	18
		S	Some event nodes removed.		18
I	Vogelaar   [25]   d   PDF	I	Dutch governance processes. Originally modeled with YAWL. Transformed to EPCs using the transformation rules from the source document.	en	81
C	Exams   e   a   exam	I	Exams of a course at a German University between 2010 and 2012.	de	78
<b>Number of all EPCs</b>					<b>2290</b>

Legend: C: Category (R: reference model, I: individual model, C: controlled modeling); S: Source (s: self-created); T: Type of source (a: analogue, d: digital); F: format of source; V: Variant (I: as-is- model, S: to-be-model), #: number of EPCs

Table 2 gives an overview of the models currently contained in the corpus. Furthermore, the category and the source of the data, the nature of the source (analogue vs. digital) and the format of the source (book, text, audio or file formats) are listed in



the tables. In addition, short descriptions, the national language (German and English) used and the number of models in the respective model collection are provided. The column “V” indicates whether it is an as-is-model or a to-be-model. Spelling corrections and the introduction of the new German spelling rules were not considered as changes. On the other side, structural changes, such as the cutting or merging of certain models were viewed as changes resulting in to be-models. Most of the changes resulted from the correction of syntactic errors, such as the correction of missing events or functions; the correction of edges only having a start or end node, or the correction of events, which due to the print version occurred only twice.

### 5.2. *Corpus Characteristics*

To characterize the model corpus, all metrics from [ME10] were used. The 33 metrics of the actual models added to the corpus are presented in table 3. A Java program was developed to calculate the metrics. This program also supports the exchange format (AML) defined in the corpus conventions. Metrics which could not be calculated, for example because of a very high computational complexity or deadlocks, were referred to by empty cells in the table. This mainly occurred in the calculation of the cross-connectivity (CC). In contrast to the computational complexity, deadlocks can possibly be resolved by further model transformations. This underlines the necessity of defining appropriate transformation rules.

Considering the complexity of the models, individual models added to the corpus tend to be larger than reference models. This can be explained by the higher level of the reference models’ abstraction on the one hand, and by the hierarchization and decomposition of reference models on the other hand. The high complexity of individual models, therefore, leads to a higher diameter (diam).

If the connectivity coefficient (CNC) is less than zero, there are more nodes than edges. The larger the coefficient of connectivity is, the more edges exist in comparison to the nodes. More edges than nodes in EPCs can be generated either by a division and reunification of the process or by loops. Thus, this metric can be seen as a first indicator of complexity. A higher connectivity is discernible when comparing reference models with individual models. This leads to a higher complexity of individual models and can be confirmed by the values of the control flow complexity (CFC). The SAP reference model describes a special feature for the R/3 System, which still has a very high CFC value - even if the CNC value is low. This is due to the fact that a high number of split connectors and an equally high number of outgoing edges, which however will no longer be merged in the further course, exist in the models added to the corpus.

Table 3: Corpus characteristics

Name	$S_{E_s}$	$S_{E_{int}}$	$S_{E_e}$	$S_E$	$S_F$	$S_{S_{AND}}$	$S_{J_{AND}}$	$S_{S_{XOR}}$	$S_{J_{XOR}}$	$S_{S_{OR}}$	$S_{J_{OR}}$	$S_C$	$S_N$	$S_A$	diam	$\Delta$	$D$	CNC	$CNC_K$	CN	$\bar{d}_c$	$d_c$	$\Pi$	$\Xi$	$\Lambda$	MM	CH	CYC	TS	CFC	JC	CP	CC
<b>Reference models</b>																																	
ECO-Integral	1,12	2,85	1,10	4,92	6,88	0,02	0,04	0,31	0,12	0,35	0,41	1,29	13,08	12,33	7,33	0,12	0,02	0,88	11,78	0,55	1,36	1,45	0,64	0,56	0,37	0,96	0,07	0,10	0,51	2,18	2,00	0,12	0,25
Retail-H 1996	2,66	12,00	2,68	17,34	10,25	1,02	1,21	3,45	3,30	0,27	0,38	9,71	37,30	40,43	22,50	0,04	0,06	1,06	43,95	4,13	3,31	4,27	0,47	0,30	1,50	4,25	0,47	0,00	1,93	11,68	11,50	0,04	0,10
Retail-H 2004	2,55	12,10	2,65	17,30	10,40	1,00	1,17	3,47	3,22	0,25	0,37	9,57	37,27	40,37	22,68	0,04	0,06	1,06	43,86	4,10	3,29	4,23	0,47	0,31	1,52	4,20	0,45	0,00	1,83	11,50	11,15	0,04	0,10
ITIL	1,00	6,00	1,00	8,00	5,00	0,00	0,00	2,00	1,00	0,00	0,00	6,00	19,00	18,00	9,00	0,05	0,00	0,95	17,05	0,00	2,00	3,00	0,82	0,50	1,00	2,00	0,00	0,00	0,00	4,00	2,00	0,06	0,07
SAP R/3	3,90	3,12	4,53	11,50	4,03	1,08	1,09	0,93	1,02	0,62	0,46	5,21	20,74	20,80	9,18	0,09	0,03	0,94	21,11	1,37	3,30	4,36	0,58	0,29	0,81	6,02	0,43	0,02	3,16	1187,98	179,48	0,09	0,17
Y-CIM 2.1	7,29	20,43	2,57	30,29	21,14	3,43	1,86	3,71	1,00	1,57	6,00	17,57	69,00	77,86		0,14	0,01	0,90	88,31	9,86	3,80	4,57	0,36	0,12	0,86	21,71	0,22		8,29	22,14	83,57	0,08	
Y-CIM 1998	1,96	3,67	1,07	6,71	5,04	0,46	0,27	0,75	0,27	0,18	0,96	2,87	14,62	14,86	8,78	0,14	0,04	0,90	15,37	1,34	2,98	3,44	0,54	0,32	0,55	3,98	0,24	0,11	0,96	3,15	26,78	0,12	0,20
Y-CIM 1994	1,96	3,64	1,09	6,69	5,02	0,46	0,27	0,73	0,27	0,18	0,96	2,87	14,58	14,80	8,62	0,14	0,04	0,90	15,31	1,37	2,97	3,42	0,54	0,32	0,55	3,95	0,24	0,11	0,96	3,11	22,13	0,12	0,20
Average	2,03	6,32	1,25	8,52	5,68	1,10	0,68	1,43	1,28	0,49	1,97	5,30	19,16	22,39	6,86	0,04	0,02	0,07	26,08	3,22	0,80	1,03	0,14	0,13	0,43	6,58	0,18	0,05	2,64	7,33	61,34	0,04	0,07
SD	2,81	7,98	2,09	12,84	8,47	0,93	0,74	1,92	1,28	0,43	1,19	6,89	28,20	29,93	12,58	0,09	0,03	0,95	32,09	2,84	2,88	3,59	0,55	0,34	0,89	5,88	0,26	0,05	2,21	155,72	42,33	0,08	0,15
<b>Individual models</b>																																	
Custom B2B	1,17	25,91	2,59	29,67	23,07	1,09	1,13	5,57	3,28	0,04	0,11	11,26	64,00	68,59	45,30	0,04	0,05	1,03	73,63	5,59	2,85	3,48	0,57	0,55	1,41	4,11	0,24	0,29	1,52	12,52	8,87	0,05	
E-Payment	1,95	36,11	1,61	39,66	31,32	1,76	1,76	6,42	6,47	0,24	0,24	16,92	87,90	95,74	44,60	0,02	0,06	1,05	104,32	8,84	3,09	3,82	0,24	0,47	2,55	1,34	0,46	0,12	2,50	16,58	17,00	0,02	0,06
Business registration	2,42	26,08	2,25	30,75	20,58	1,13	0,83	6,08	5,46	0,17	0,17	13,83	65,17	71,75	38,79	0,02	0,07	1,08	79,14	7,58	3,28	4,58	0,43	0,44	1,92	3,58	0,37	0,18	2,04	15,83	15,29	0,02	
GK-Rewe	1,67	12,92	1,00	15,23	6,97	2,36	2,54	1,39	1,31	0,85	0,85	10,77	32,97	38,74	21,05	0,04	0,14	1,17	45,94	7,29	3,35	5,08	0,37	0,21	1,69	1,64	0,58	0,13	6,18	16,08	17,85	0,05	0,18
PMC	1,00	27,83	1,11	29,94	28,83	1,50	1,06	5,50	4,39	0,22	0,39	13,06	71,83	79,78	49,33	0,02	0,08	1,10	88,68	8,94	3,35	5,06	0,21	0,48	2,17	5,72	0,35	0,04	3,11	25,39	35,50	0,02	0,06
Vogelaar	1,00	30,16	1,00	32,16	31,14	1,58	1,53	5,62	5,83	0,37	0,37	15,30	78,59	87,16	52,53	0,02	0,07	1,10	96,72	9,57	3,18	4,04	0,18	0,49	3,06	0,70	0,48	0,07	3,09	15,53	15,70	0,02	0,02
Average	0,58	7,65	0,69	7,95	9,27	0,47	0,62	1,85	1,91	0,28	0,27	2,35	18,78	19,78	11,24	0,01	0,03	0,05	20,67	1,45	0,19	0,67	0,15	0,12	0,60	1,93	0,12	0,09	1,64	4,36	8,97	0,02	0,07
SD	1,53	26,50	1,59	29,57	23,65	1,57	1,48	5,10	4,46	0,31	0,35	13,52	66,74	73,63	41,93	0,03	0,08	1,09	81,40	7,97	3,18	4,34	0,33	0,44	2,13	2,85	0,41	0,14	3,07	16,99	18,37	0,03	0,08
<b>Models from controlled modeling scenarios</b>																																	
Exams	1,26	13,89	2,91	18,05	14,37	0,62	0,51	3,94	1,82	0,27	0,39	7,64	40,06	42,03	23,58	0,03	0,06	1,04	44,30	3,53	3,06	3,39	0,45	0,45	1,54	6,49	0,30	0,19	1,00	10,10	6,50	0,03	0,08

Legend:  $S_{E_s}$ : number start events;  $S_{E_{int}}$ : number internal events;  $S_{E_e}$ : number end events;  $S_E$ : number events;  $S_F$ : number functions;  $S_{S_{AND}}$ : number AND splits;  $S_{J_{AND}}$ : number AND joins;  $S_{S_{XOR}}$ : number XOR splits;  $S_{J_{XOR}}$ : number XOR joins;  $S_{S_{OR}}$ : number OR splits;  $S_{J_{OR}}$ : number OR joins;  $S_C$ : number connectors;  $S_N$ : number nodes;  $S_A$ : number edges; diam: diameter;  $\Delta$ : density(1);  $D$ : density(2); CNC: coefficient of connectivity;  $CNC_K$ : coefficient of network complexity; CN: cyclomatic number;  $\bar{d}_c$ : avg. connector degree;  $d_c$ : max. connector degree;  $\Pi$ : separability;  $\Xi$ : sequentiality;  $\Lambda$ : depth; MM: mismatch; CH: heterogeneity; CYC: cyclicity; TS: token splits; CFC: control flow complexity; JC: join complexity; CP: weighted coupling; CC: cross-connectivity; SD.: standard deviation

The cyclicity value (CYC) covers the ratio of the number of nodes in loops to the number of all nodes in a model. Considering these values, a trend of individual models to a higher value than the reference models can be observed. Especially the Retail-H reference model with a value of 0 is demonstrative for that. This is due to the fact of hierarchization and decomposition of reference models, which, in most cases, is not done in individual models. Therefore, loops could occur without impact on the CYC value. Against that background, the argumentation of a higher cyclicity at individual models would be false. In fact, the metric is not sensible for that aspect. If hierarchized and decomposed models were transferred into a flat EPC, these models would contain cycles, too. However, transformations like this are affiliated with several challenges [26].

In addition, the mean values of the metrics for the reference and individual models were calculated. A comparison of these values provides information on the general properties of the considered models. They show, for example, that individual models are, on average, twice as large as reference models. An exception is the mean of the OR connectors, which are used in individual models only half often as in reference models. The CYC and diam-values follow a similar relationship, while the CNC and  $d_C$  values show no significant differences between the two categories.

In order to analyse the stability of these similarity metrics, the standard deviations for the reference and individual models were calculated. Table 3 shows that these values do not vary over the corpus very much. However, in many cases, the values of individual models are higher. An exception can be observed at the control flow complexity (CFC) with a high standard deviation of about 155. This is due to the SAP R/3 reference model and indicates a superior complex structure of it. Ignoring that reference model at calculating the CFC value, it would be about 13.

The application scenario provides information about the different corpus characteristics on the one hand and shows its ability in context of metrics evaluation on the other hand. It would be very hard or even impossible to discover specific aspects while evaluating techniques, methods and algorithms without a substantial data basis, which could then lead to an inappropriate or defective interpretation of research results. The example also emphasizes the need for consideration of the principle of powerfulness while developing a model corpus.

## 6. Discussion and Outlook

The presented model corpus was developed based on a procedure model. In order to lend a certain basic width to the model corpus, different model collections, such as reference models, individual models as well as models from controlled modelling scenarios, were selected. Both analogue and digital sources were considered. Except for ITIL, various changes were made for all reference models, providing the to-be models. These changes primarily concern the adjustment of syntactic errors identified manually or the transformation of constructs as well as double connectors or sequence operators. Since the as-is-models and the to-be-models were added to the corpus with respect to different conventions, different constructs and syntactical rules are used in these models. The simultaneous existence of the as-is models and as the to-be models allows a wide range for application scenarios.

Moreover, the model corpus was extended by further model collections. Different national languages were considered (Y-CIM, SAP-R/3), some model collections were

taken from various sources (SAP-R/3) as well as from different years of publication (Retail-H, Y-CIM).

Altogether, the model corpus consists of 16 model collections with 2290 EPCs. For eight of these collections, to-be models have been created. The ARIS export data format AML was chosen as the unified exchange format of the model corpus. Generally, in contrast to a (simple) single model, the presented model corpus provides several models of different domains, sizes and national languages. Nevertheless, the developed model corpus is narrow in size in comparison to the whole domain. Thus, the developed corpus cannot be seen as representative. This can be drawn back to the availability of free accessible models. However, the model corpus can be used in a wide range of application scenarios. In order to stress the applicability, one example application scenario was presented in this paper. Within that scenario, the model corpus was characterized by the use of 33 metrics provided by Melcher [ME10]. Thus, the authors have taken a first step towards the realization of the presented vision of an extensive model corpus. In contrast to existing approaches, the scientific need for concrete digitally processable models has been addressed, since in many cases a lack of a uniform data basis exists.

The addressed complexity of the developed model corpus enables both the evaluation of existing algorithms, methods and techniques and their (further) development. Here, some possible application scenarios have been outlined briefly, which should be investigated in more detail in future work.

In addition to the application scenarios, the continuous development of the model corpus by adding further models (even by other researchers) is in the focus of further work. Moreover, the licensing issues that are associated with the provision of the model corpus have to be resolved. Only then, the initially formulated principle of openness can be worn fully into account.

## References

- [1] Dijkman, R., et al., Similarity of business process models: Metrics and evaluation. *Information Systems*, 2011. 36 (2): p. 498-516.
- [2] Houy, C., et al. Business Process Management in the Large. in *Business & Information Systems Engineering*. 2011.
- [3] Mendling, J., Metrics for process models : empirical foundations of verification, error prediction, and guidelines for correctness. 2008: Springer.
- [4] Fettke, P. and P. Loos, Zur Identifikation von Strukturanalogien in Datenmodellen – Ein Verfahren und seine Anwendung am Beispiel des Y-CIM-Referenzmodells von Scheer. *Wirtschaftsinformatik*, 2005. 47 (2): p. 89-100.
- [5] Ekanayake, C., et al., Approximate Clone Detection in Repositories of Business Process Models, in *Business Process Management*, A. Barros, A. Gal, and E. Kindler, Editors. 2012, Springer Berlin Heidelberg. p. 302-318.
- [6] Walter, J., P. Fettke, and P. Loos. Zur Identifikation von Strukturanalogien in Prozessmodellen. in *Tagungsband der Multikonferenz Wirtschaftsinformatik (MKWI 2012)*. 2012. Braunschweig, Germany.
- [7] Ardalani, P., et al. Towards a Minimal Cost of Change Approach for Inductive Reference Model Development. in *Proceedings of the 21st European Conference on Information Systems (ECIS 2013)*. 2013. Utrecht, Netherlands: AIS.
- [8] Koch, S., S. Strecker, and U. Frank, Conceptual Modelling as a New Entry in the Bazaar: The Open Model Approach, in *Open Source Systems*, IFIP 203, E. Damiani, et al., Editors. 2006, Springer: Berlin. p. 9-20.
- [9] Fellbaum, C., et al. WordNet: An Electronic Lexical Database. 1998 27.10.2010 [cited 2010 15.11.2010]; Available from: <http://wordnet.princeton.edu/>.

- [10] Kunze, C., Semantische Relationstypen in GermaNet, in *Semantik im Lexikon*, S. Langer and D. Schnorbusch, Editors. 2005, Narr. p. 161-178.
- [11] Fettke, P. and P. Loos, *Der Referenzmodellkatalog als Instrument des Wissensmanagements - Methodik und Anwendung*, in *Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung*, J. Becker and R. Knackstedt, Editors. 2002, Springer: Berlin et al. p. 3-24.
- [12] Cayoglu, U., et al. The Process Model Matching Contest 2013. in *4th International Workshop on Process Model Collections: Management and Reuse (PMC-MR'13)*. 2013. Beijing.
- [13] Krcmar, H., et al., eds. *Informationssysteme für das Umweltmanagement - Das Referenzmodell ECO-Integral*. 2000, Oldenbourg: München, Wien.
- [14] Becker, J. and R. Schütte, *Handelsinformationssysteme*. 1996, Landsberg/Lech: verlag moderne industrie.
- [15] Becker, J. and R. Schütte, *Handelsinformationssysteme. Domänenorientierte Einführung in die Wirtschaftsinformatik*. 2. ed. 2004, Frankfurt am Main: Redline Wirtschaft.
- [16] *Office of Government Commerce, ITIL - Service Strategy*. 2010, Norwich: TSO Information & Publishing Solutions.
- [17] *Office of Government Commerce, ITIL - Service Design*. 2010, Norwich: TSO Information & Publishing Solutions.
- [18] *Office of Government Commerce, ITIL - Service Operation*. 2010, Norwich: TSO Information & Publishing Solutions.
- [19] *Office of Government Commerce, ITIL - Service Transition*. 2010, Norwich: TSO Information & Publishing Solutions.
- [20] *Office of Government Commerce, ITIL - Continual Service Improvement*. 2010, Norwich: TSO Information & Publishing Solutions.
- [21] Keller, G. and T. Teufel, *SAP R/3 prozeßorientiert anwenden – Iteratives Prozeß-Prototyping zur Bildung von Wertschöpfungsketten*. 1998, Bonn et al.: Addison-Wesley.
- [22] Scheer, A.-W., *Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse*. 2. ed. 1998, Berlin et al.: Springer.
- [23] Scheer, A.-W., *Business Process Engineering - Reference Models for Industrial Enterprises*. 2. ed. 1994, Berlin et al.: Springer.
- [24] Kahlert, D. *Grundkurs Rechnungswesen*. 2010 [cited 2010 23.11.2010]; Available from: <http://www.tu-chemnitz.de/wirtschaft/sapr3/gkrewe/epk/>.
- [25] Vogelaar, J.J.C.L., et al., Comparing Business Processes to Determine the Feasibility of Configurable Models: A Case Study, in *Business Process Management Workshops, LNBIP 100*, F. Daniel, K. Barkaoui, and S. Dustdar, Editors. 2012, Springer: Berlin. p. 50-61.
- [26] Walter, J., P. Fettke, and P. Loos. Decomposition and Hierarchization of EPCs: A Case Study. in *The 4th Int. Workshop on Process Model Collections: Management and Reuse (PMC-MR 2013)*. 2013. Beijing, China.

# Context-Sensitive Framework for Visual Analytics in Energy Production from Biomass

Pekka WARTIAINEN <sup>a,1</sup>, Anneli HEIMBÜRGER <sup>a</sup> and Tommi KÄRKKÄINEN <sup>a</sup>

<sup>a</sup> *Department of Mathematical Information Technology, University of Jyväskylä, Finland*

**Abstract** Data masses require a lot of data processing. Data mining is the traditional way to convert data into knowledge. In visual analytics, humans are integrated into the process as there is continuous interaction between the analyst and the analysis software. Data mining methods can be utilized also in visual analytics where the priority is given to the visualization of the information and to dimension reduction. However, the provided data is not always enough. There is a large amount of background contextual information, which should be included into the automated process. This paper describes a context-sensitive approach, in which we utilize visual analytics by studying all phases in the process according to our "sensing, processing and actuation" framework. Experimental studies show that our framework can be very useful in the process of analyzing causes for and relations between variable changes with laboratory-scale power plant data.

**Keywords.** Visual Analytics, Context, Context Sensitive, Energy Production, Visualization

## Introduction

While analyzing industrial processes, especially those of energy production, the context information plays a significant role in acquiring reliable results. High level computational methods are mandatory for processing data, but, if the context is not defined, results are not fully understood. Energy production from biomass has been a challenging area due the organic compounds in the biomass [19]. Among other things, gas from burning biomass forms chloride acids in high temperatures. Another important factor to be taken into account when utilizing biomass is the amount of small particulates produced.

Visual Analytics is a relatively new research field – the first book in the field was published in 2005 [21]. It refers to interactive methods and technologies that could be applied for presenting the results of data mining process to users [21]. Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex datasets [12].

---

<sup>1</sup>Corresponding Author: Pekka Wartiainen, Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora) FI-40014 University of Jyväskylä, Finland; E-mail: pekka.wartiainen@ju.fi.

Industrial process monitoring is a well-suited genre for applications of visual analytics due its elements of data analysis and visualization. However, the set of available examples is scarce. One possible reason for this is that visual analytics requires a lot from software methods and algorithms: user interaction, data analysis and visualization methods are too far apart from each other [12]. In visual analytics, these methods should be used simultaneously without restrictions imposed on them in some of these areas. Highly interactive interfaces combined with automated data analysis and visualization will reduce the gap between the user and the computer. Interaction possibilities are not limited only to parameter tuning and data exploration: also contextual information can be made available.

The concept of contextual sensing fits well also to the idea of visual analytics. In visual analytics, the researcher is seen as a part of knowledge mining process with his/her background knowledge [12]. That background knowledge is important for deeper understanding of the problem and helpful in finding more reliable solutions. In the same way, sensing of contextual facts provides more information, which in this case is necessary for a proper and valid analysis.

Based on this, if something changes in the context, it will have direct effect to the process and measurements. Change-point detection addresses the problem of discovering time points at which properties of time-series data change [11]. There are numerous ways of implementing change-point detection algorithms, but traditionally they are all based on statistical methods and properties [20]. In this paper, we introduce a context-sensitive framework and an implementation of it, which utilizes context-sensitive approach and change-point detection methods for visual analytics.

Our paper is organized as follows. Related work is discussed in Section 1. The principles of our iterative context-sensitive framework are introduced in Section 2. In Section 3, we present the preliminary implementation of the framework. Conclusions will be given in Section 4.

## 1. Background

Although visual analytics and data mining have been extensively researched, there are still many challenges [22]. Huge data sets and data bases require enormous amounts of storing capacity and almost incomprehensibly fast data transfer connections. In many application areas, data is complex or inconsistent. Also, it may happen that even if there is a huge amount of data with many variables, there is still very little data that is suitable for training [22]. Therefore, handling big data is challenging and finding the optimal solution for feature selection and evaluation of results is difficult. Existing visual analytics software and frameworks are still most likely related to certain application fields [1, 17].

Context-sensitivity in the field of computing can be defined, e.g. as in [8], thus: "A computational method, a computer system, or an application is context-sensitive if it includes context-based functions and if it uses context to provide relevant information and services to the user, where relevancy depends on the user's situation". Regarding industrial applications, one could add that context-sensitivity depends not only on the user's situation but also on the stage of the process and the state of the environment.

Context awareness is often encountered in telecommunications and web-page document analysis where location and other sensor measurement information is part of the

data [2, 18]. System is context-aware if it "uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task" [3]. Architecture of context-aware applications is presented in [7]. In our work, we prefer using the term context-sensitive approach, since it gives more choices and sets fewer limitations.

Traditionally, change-point detection uses statistical methods for finding fluctuation in the data. Recently, there was an effort to use anomaly detection in data mining to detect change-points. By defining an anomalous pattern as one "whose frequency of occurrences differs substantially from that expected, given previously seen data", Keogh et. al. presented a way where anomalies are not explicitly formulated [14]. Otherwise, fault detection and pattern finding related to industrial purposes have been researched a lot [5, 9–11, 13].

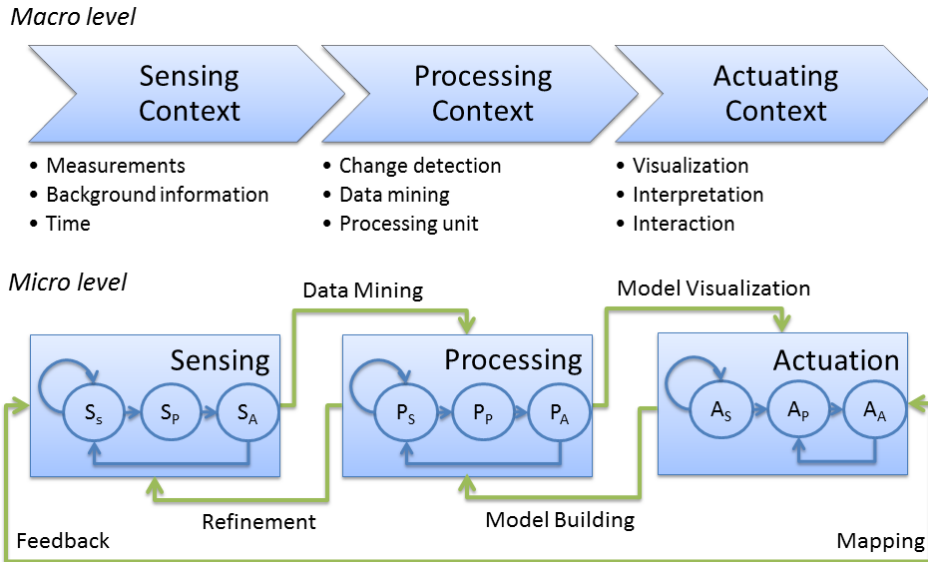
## 2. Iterative Context-sensitive Approach

In industrial processes, contextual information contains all meta data related to measurement environment, software architecture, data processing and visualization and also to preliminary knowledge from the related application area. In a complete analysis chain, contextual information should be taken into account in every analysis phase and should be included as a part of automated processing. An industrial application, which for example uses burning biomass to produce heat and electricity, has a very specific context basis. Therefore, in order to compute reliable results, different context environments are required in analyses for different industrial applications.

To meet these challenges, we are extending the EJC2012 context-sensitive SPA framework [23] with the iterative SPA approach (Figure 1). The iterative SPA framework is divided into two hierarchical levels. On general macro level, the energy production process is analyzed as a straight-forward analysis process, from variable measurements to visualization and interpretation of results. Different contexts are related to each SPA step which are considered in the overall analysis process. For example, the sensing context includes information about how and where measurements are taken, processing context describes software analysis methods, and actuating context defines the kind of visualization used.

From the visual analytics point of view, this straight-forward macro level is not enough to provide holistic and accurate information view on user. The model of visual analytics allows the process move between automated processing and visualizations while also mapping the raw data for visualizations [12]. Hence, we introduce an iterative micro level with the SPA structure. The main additional benefit of this is the possibility to move back and forth and jump over the SPA steps. Micro level is defined below macro level in the hierarchy and contains all the same contextual elements. In addition, each SPA step is divided into iterative sub-phases  $S_{\{S,P,A\}}$ ,  $P_{\{S,P,A\}}$ , and  $A_{\{S,P,A\}}$  that follow the SPA structure. Here, a sub-process contains one set of iterative SPA sub-phases. In general, sensing sub-phase  $\{S, P, A\}_s$  includes an update loop that initializes a new set of parameters and restarts the sub-process. A more detailed description of the SPA phases and their sub-phases is given in the following chapters.





**Figure 1.** The framework of iterative SPA with elements of visual analytics. On macro level, the analysis process is straight forward from measurements to analysis. On micro level, there is a more detailed structure in the analysis chain, where each process phase contains also iterative sub-phases. The next phase can be chosen by the type of the action required. For example, starting from the **Sensing** phase, the user can do *data mining* by advancing to the **Processing** phase or *mapping* and visualize raw data directly in the **Actuation** phase.

### 2.1. Sensing contextual information

Sensing starts from inquiring all possible data related to the measurement environment and equipment as well as background knowledge from application. Also, when conducting an experiment, everything should be logged as well as possible, especially if adjustments are made. This is the main contribution in sub-phase  $S_S$ . The more information is gathered the better. Irrelevant information can be automatically reduced afterwards.

Once the experiment has been completed, all measured data and context information is processed into variables (subphase  $S_P$ ). While creating new variables from context information, one has to decide whether to use them in computation. In general, quantitative or continuous variables can be used in computation, but qualitative or categorical variables provide meta-data for researcher for the interpretation and evaluation of results and later for decision making. Variables are also classified as input- or output-type variables according to their function. For example, temperature and pressure measurements are output-type (monitored) variables while fuel feeding and bottom coarse conveyor belt are input-type (controlling) variables.

In the last sub-phase  $S_A$ , the next actions are decided. In dynamic industry environment, e.g. in a power plant, context environment may change during time. For example, it is possible to change the type or quality of fuel. A radical change, like that of fuel in the context environment, affects the rest of the analysis chain. While analyzing the efficiency of a certain fuel type, a model of parameter settings is built. Due the some elemental facts, different fuels behave differently, and the original model will fail to produce reliable results. If we find a change in the context environment, the process model should be updated accordingly.

## 2.2. Processing Context

Changes in the context environment have to be adapted in the processing phase. Sensing these changes can be done manually or automatically, depending on the case in phase  $P_S$ . Possible changes could be triggered by a different fuel type or an abnormal behavior of the signals. As a response to these changes, the processing model and parameters are updated, but also going back to sensing phase is possible.

Processing phase is computationally heavy since all software computations are done in this phase (more specifically, step  $P_P$ ). However, with modern computers, computations are relatively fast with small and large data sets. Here, a large data set contains more than 100 variables, with several thousands of measurements. In our framework, computations will include the steps of preprocessing (e.g. synchronization), transformation (e.g. dimension reduction) and change-point detection.

In the last phase  $P_A$ , the results of processing are briefly verified. For example, the values for each change-point detector are checked and detection thresholds are fine-tuned. A more explicit explanation of detectors and threshold is given in Section 3. Based on human decision, the analysis process may be continued to Actuation phase or iterated back to phase  $P_S$  if some tuning of parameters for computation is needed.

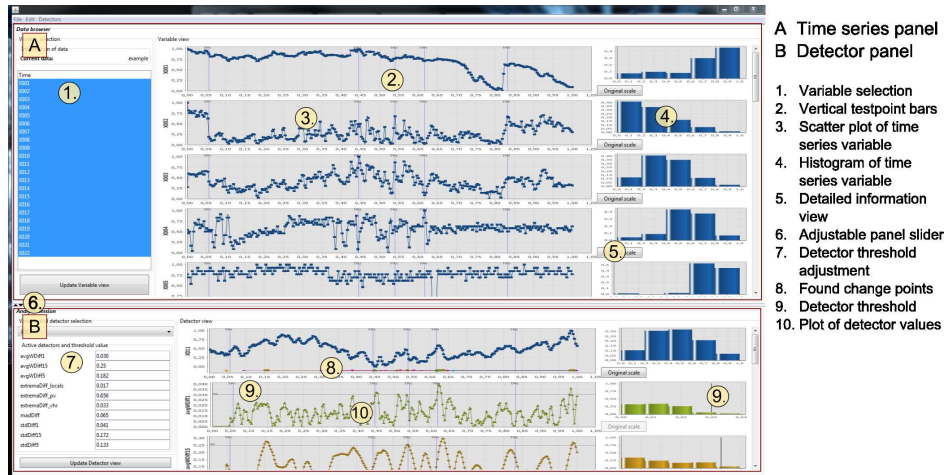
## 2.3. Actuation context in visual framework

Actuation phase starts with (sub-phase  $A_S$ ) sensing the context at the user's end. Different contexts, e.g. different user roles, will be selected according to current situation. In the ultimate scenario, these contexts could be selected automatically based on the knowledge gathered from the user. For example, different factors could be formed of the role of the user, expertise of the working group, etc. In practice, an overall view is presented automatically, but the user may find more detailed information of the data when needed.

After the initialization of the user interface, the results of computations are visualized and examined (sub-phase  $A_P$ ). Often, the cases with industrial data require discussions and interpretation. Validation of the results, with data from real world, is always a huge challenge. One of the best validation methods (perhaps the only) is to trust experts' opinion. In sub-phase  $A_A$ , decisions are made based on acquired knowledge. If the results are insufficient, the user can go back to the previous SPA steps and for example adjust the parameters or use different methods. On the other hand, if the results are proven reliable, measured information has been transformed to some knowledge which hopefully solves the research problem.

## 3. Analysis framework for effective computation and visualization

Motivated by the lack of contextual support in existing applications, we started the development of a visual analytics software for analyzing time-series data. Matlab [15] was chosen as a base of the framework because of its computational features. Matlab has a collection of implemented algorithms, and development with it is often faster than with traditional languages. However, building a GUI in Matlab is a challenge because of visual analytics requirements for user interactions. Also, it is known that after a version upgrade in Matlab all features may not function properly in old interfaces. Therefore Java was



**Figure 2.** Java GUI with example data. In the upper panel, time-series data is visualized. Change points are analyzed on the lower panel.

chosen as the language for GUI development. A Java GUI was built using Java's Swing library [4], and visualizations utilize the JFreeChart library [6]. The main advantage of using Java GUI on top of Matlab is that all Java visualization tools can be combined into powerful data processing techniques in Matlab.

The GUI (Figure 2) is divided into two main panels which are aligned horizontally. The upper panel, *Data browser*, offers tools for analyzing time-series visually, and the lower panel, *Analysis session*, provides an interface to apply change-point detection and to adjust detector values. In a basic workflow, the user selects variables for visual inspection in the *Data browser* and then, in the *Analysis session*, a variable for further analysis, e.g. change-point detection. Variables in scatter plots are normalized between 0 and 1, and the plots are synchronized in time for easier comparison.

After change-point detection, the findings can be exported and loaded back into the system as a new data set and the results can be inspected again on the *Data browser*. In this phase, the crucial feature for finding relations between variables is the ability to mark interesting points in time as test points, which are plotted on each variable (Number 2 in Figure 2). Now the user may find similarly behaving variables before and after any test point. Of course, the automatically computed similarities are given to the user, but an expert's opinion is required to achieve reliable results with real world data.

In the experiments, we found out that the key element in mining reliable results is the change-point detection. Change-point detection is operated in the lower panel *Analysis session* of the GUI. On the left, one variable can be selected and then plotted on the top graph of the lower panel. The rest of the space is reserved for detector plots, where the results of different change-point methods are visualized. Each detector has a horizontal threshold bar (Number 9 in Figure 2). All detector values exceeding the threshold level are considered as change points. The user can fine-tune a threshold value for each detector separately, based on his/her expertise on the field. Detected change points can be drawn into the top plot with the corresponding variable.

Color design in graphical user interfaces plays an important role in a successful data analysis and understanding process. Colors can be used for advantage by highlighting

correct information, but with careless usage of colors the whole interface may become unusable. In this framework, general guidelines of GUI design have been followed. The color design of scatter plots is chosen with a color palette toolkit provided by NASA's Ames Research Center [16].

The GUI is also scalable in the contextual sense. The big picture from data can be seen on the main window, but more detailed information is available. For example, in the original space window (opened with the *Original scale* button) the requested variable is plotted in a different context and with its original values and time scale containing more specific information about the measurements and the change-points. Thus the researcher can find more relevant information of the interesting points in time.

Considering the iterative SPA framework, the GUI concentrates mainly to processing and actuation phases. The data visualized in the upper panel of the GUI supports actuation and decision making in every SPA sub-phase. The lower panel implements the processing phase with visualizations computed by automated methods in Matlab. However, the sensing phase contains still a few steps, which are conducted manually in Matlab, for example, adding context information into the system.

#### 4. Conclusions

In case of a power plant, the idea of visual analytics is to give tools for research and development tasks rather than to build a fast controlling system. We have done to this by extending the context-sensitive SPA framework with an iterative structure. By offering ways to loop back and provide feedback to previous steps, iterative SPA framework facilitates getting deeper understanding of measured information.

In order to utilize the iterative SPA framework, we started developing a GUI for Matlab, which integrates an interactive user interface and visualizations to powerful automated computations. The development work is still in progress with analysis methods and to make context information more transparent. However, preliminary work with real world data provided by Valmet (previously called Metso Power) has given positive results. Our novel context-sensitive approach and framework, including the analysis-chain, have given new insights and ways to data analysis. Our tests show that the change-point detection methods have the key role in achieving reliable results.

In the future, we would like to concentrate our efforts on creating a massive library of change-point methods. This versatile library would provide tools for different tasks, as no particular method is the best in every situation. The second improvement to our framework is not to deal only with the context of a single point in time but with the context of a region of interest (ROI). In industrial applications, changes might be slow and take for example half hour to complete. During that time it is impossible to set a single point in time for the change, but it is happening in a region of time points.

#### Acknowledgements

The authors thank the OSER-project funded by European Social Fund. Also, our thanks to Valmet, which has been providing material for the research.

## References

- [1] José A. Castellanos-Garzón, Carlos Armando García, Paulo Novais, and Fernando Díaz. A visual analytics framework for cluster analysis of dna microarray data. *Expert Syst. Appl.*, 40(2):758–774, 2013.
- [2] Stefano Ceri, Florian Daniel, Maristella Matera, and Federico M. Facca. Model-driven development of context-aware web applications. *ACM Trans. Internet Technol.*, 7(1), February 2007.
- [3] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, January 2001.
- [4] Robert Eckstein, Marc Loy, and Dave Wood. *Java Swing*. O’Reilly, Beijing, 2. edition, 2002.
- [5] Ada Wai-chee Fu, Eamonn Keogh, Leo Yung Hang Lau, and Chotirat Ann Ratanamahatana. Scaling and time warping in time series querying. In *Proceedings of the 31st international conference on Very large data bases*, VLDB ’05, pages 649–660. VLDB Endowment, 2005.
- [6] David Gilbert. Jfreechart. <http://www.jfree.org/>, 2000.
- [7] Anneli Heimbürger, Yasushi Kiyoki, Tommi Kärkkäinen, Ekaterina Gilman, Kyoung-Sook Kim, and Naofumi Yoshida. On context modelling in systems and applications development. In *Proceedings of the 2011 Conference on Information Modelling and Knowledge Bases XXII*, pages 396–412, Amsterdam, The Netherlands, The Netherlands, 2011. IOS Press.
- [8] Anneli Heimbürger, Miika Nurminen, Teijo Venäläinen, and Suna Kinnunen. Modelling contexts in cross-cultural communication environments. In *Proceedings of the 2011 Conference on Information Modelling and Knowledge Bases XXII*, pages 301–311, Amsterdam, The Netherlands, The Netherlands, 2011. IOS Press.
- [9] Chun-Chin Hsu, Mu-Chen Chen, and Long-Sheng Chen. Intelligent ica-svm fault detector for non-gaussian multivariate process monitoring. *Expert Syst. Appl.*, 37(4):3264–3273, April 2010.
- [10] Chun-Chin Hsu, Mu-Chen Chen, and Long-Sheng Chen. A novel process monitoring approach with dynamic independent component analysis. *Control Engineering Practice*, 18(3):242 – 253, 2010.
- [11] Yoshinobu Kawahara. Change-point detection in time-series data by direct density-ratio estimation. *Direct*, 4(2):389–400, 2009.
- [12] Daniel A. Keim, Joern Kohlhammer, Geoffrey Ellis, and Florian Mansmann, editors. *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, November 2010.
- [13] Eamonn Keogh and Shruuti Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining Knowledge Discovery*, 7(4):349–371, October 2003.
- [14] Eamonn Keogh, Stefano Lonardi, and Bill ’Yuan-chi’ Chiu. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD ’02, pages 550–556, New York, NY, USA, 2002. ACM.
- [15] MATLAB. *version 7.11.0.584 (R2010b)*. The MathWorks Inc., 2010.
- [16] Ames Research Center NASA. Using color in information display graphics. <http://colorusage.arc.nasa.gov/>.
- [17] Kristien Ooms, Gennady L. Andrienko, Natalia V. Andrienko, Philippe De Maeyer, and Veerle Fack. Analysing the spatial dimension of eye movement data using a visual analytic approach. *Expert Syst. Appl.*, 39(1):1324–1332, 2012.
- [18] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90, 1994.
- [19] Jaani Silvennoinen and Merja Hedman. Co-firing of agricultural fuels in a full-scale fluidized bed boiler. *Fuel Processing Technology*, 2011.
- [20] Silvio Simani and Ronald J. Patton. Neural networks for fault diagnosis of industrial plants at different working points. In Michel Verleysen, editor, *ESANN*, pages 495–500, 2002.
- [21] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.
- [22] Tatiana von Landesberger, Sebastian Bremm, Matthias Kirschner, Stefan Wesarg, and Arjan Kuijper. Visual analytics for model-based medical image segmentation: Opportunities and challenges. *Expert Systems with Applications*, 40(12):4934 – 4943, 2013.
- [23] Pekka Warttinen, Tommi Kärkkäinen, Anneli Heimbürger, and Sami Äyrämö. Context-sensitive approach to dynamic visual analytics of energy production processes. In Yasushi Kiyoki and Takehiro Tokuda, editors, *22th European-Japanese Conference on Information Modelling and Knowledge Bases*. MATFYZPRESS - Univerzity Karlovy, 2012.

# Linguistic rules for automatic summarization of spoken meetings

Nils WEBER <sup>a,1</sup>, Christoph EIGENSTETTER <sup>a</sup>, Antje DÜSTERHÖFT <sup>a</sup> and Markus BERG <sup>a</sup>

<sup>a</sup> *Faculty of Engineering, University of Wismar, Germany*

**Abstract.** The research project "Autoprot" intents to create a system, that automatically records meetings, subsequently transforms the spoken language into text and finally reduces this text to the most important headwords. One important problem in this process is the automatic summarization of natural language dialogues. This article describes our current work and focuses on general text mining aspects as well as on methods to detect relevant text parts in transcripts of naturally spoken meetings introducing noun chains.

**Keywords.** natural language dialogues, text mining, automatic summarization, POS-Tagging, MALT-Parsing, noun chains

## Introduction

Transforming automatically recorded audio data received from a meeting into structured meeting protocols is one of the biggest visions of natural language processing. Text understanding and mining as well as knowledge discovery are the main challenges in this context. The "Autoprot" project records meetings, subsequently transforms the spoken language into text and finally reduces this text to the most important headwords. Actually the system acts like a human transcript writer. One important problem in this scenario is the automatic summarization of text. Thus, the system extracts the meaningful parts of speech in a given context or context-free. The term of "relevant text" depends mainly on the user's perspective but there are also common aspects, which are useful during the analysis. To get a maximum of flexibility, the system provides options for an extensive parameterization to consider the individual world view of the user.

## 1. Related Work

Current work in psycholinguistics ([1], [2], [3]) shows that syntactic and semantic aspects of natural language can be derived from the usage and statistics of text data. We focus on analyzing text data from spoken meetings. Thus, we are using a more practical approach, which is usable in common office scenarios. So far we did not find an existing solution which fully complies with those requirements. Extensive work has been

---

<sup>1</sup>Corresponding Author: Nils Weber, University of Wismar; E-mail: nils.weber@hs-wismar.de

done on text summarization and headword generation in [9] and a prototype was created which achieved good results compared to existing solutions like Copernic [13], Open Text Summarizer [16] and Intellexer Summarizer [10]. Additionally there are many theoretical approaches for automatic text summarization. Gabriel et al. evaluated different approaches to summarize automatic transcripts of meeting recordings [8], including Maximal Marginal Relevance (MMR), Latent Semantic Analysis (LSA) and feature-based approaches with the conclusion, that MMR and LSA were comparable to each other but outperformed the feature-based techniques.

## 2. Linguistic approaches

In text mining there are two different ways to look at documents. The first one is the lexical approach where central objects are known words from dictionaries that are used to indicate relations between particular parts of text. The second way is the interpretation of grammatical information that is included in the text. Both methods can be combined to filter special key aspects and to improve the results.

### 2.1. Spoken language characteristics

When processing natural language dialogues, rules are more individual and can vary strongly depending on the speakers and the situation. The structure of a dialogue does not conform to a set of rules from standard language models for formal language. A big amount of the available text is used to channel the talk, there are people that function as moderators and transcripts of spoken words frequently contain a lot of constructs that impede the recognition process. The relation between two statements or the answer to a particular question is not necessarily bound to the initial question and can cover different aspects of the talk partner's ideas. Additionally common knowledge, which is not spoken, can be referenced and repetitions have to be filtered before analysis. Some characteristic features of spoken language are: a) delays, b) repetitions, c) non words, d) grammatical compliance and e) indirect correlation. Even though we assume a perfect automatic transcription of a meeting recording, we will run into problems during the interpretation process due to the aspects mentioned above. The transcript will be unstructured and without punctuation, thus we are missing a lot of information.

### 2.2. POS-Tagging and MALT-Parsing

*POS-Tagging* Before deeper analysis the text is enriched with grammatical metadata. Our first step is a Part-of-Speech-Tagging (POS) which assigns possible word types to each word of the full text. There are no official standards for abbreviations of the word types but at least there are common tag sets which can be chosen. This paper refers to the definitions of the "Stylebook for the Tübingen Treebank of Written German" [11] (see figure 1). The tools used are working on extracts of this tag set and are implemented proprietary. The underlying models are generated from extensive corpora like the TIGER Corpus [14], the BROWN Corpus [12] and customized corpora. There is a strong dependency between the used corpora and the input text. For an optimized analysis of spoken language, customized corpora for the given input type should be used (see section 2.5). In the prototype existing models for particular parsers in german language were used

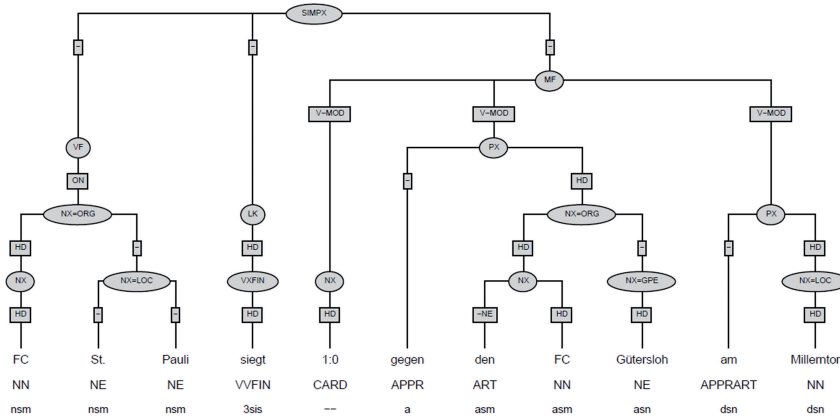


Figure 1. POS-Tagging and MALT-Parsing Results[11]

and the results of our POS-Tagging do not suffer significantly from the corpus characteristics. Special features of spoken dialogues are influencing the selection of detection rules. Those are: a) natural language, b) appearance of different speakers, c) initiative (question, answer), d) correction and negations. Correction and negation are also partially present in formal texts, but the way they are used differs. Thus, different rules for processing of spoken dialogues must be applied.

*MALT-Parsing* In addition to POS-Tagging, the MALT-Parser will return a dependency tree (structural information on the phrase level). We are using the node labels as indicators for deeper search with more advanced rules. The current prototype generates a dependency tree and specific rules to select nodes, which are enclosing significant content to generate a headword. That way, we can extract noun phrases, verbal phrases and prepositional phrases from the full text and summarize their content. The definition of those specific rules is subject of current research. Additionally the system uses the MALT-Parsing results to display the phrase structure as a tree diagram. This helps a lot to understand the connection between important text parts and word types and shows which nodes could indicate relevance.

### 2.3. Noun chains

In our prototype we are introducing different approaches to reach a deeper level of understanding. With the grammatical information from the POS-Tagging it is possible to search for typical structural patterns which are indicating significance or insignificance. We are using a strategy of "Noun chains" to choose candidates. The indicators for a deeper test are the amount and the density of nouns and proper nouns that are initially used to keep related units of text together. The length of the selected parts is adjustable and can vary strongly according to the type of text.

$$A = \text{all available word types} \quad (1)$$

$$B = \{a; b\} : \text{positive indicators (a = noun, b = proper noun)} \quad (2)$$



$$C = \{x \in A \cap x \notin B\} : \text{negative indicators} \quad (3)$$

Our testing algorithms are working in two directions. First we choose an element of set B which represents the center of the analysis. Subsequently the words between this starting point and the next positive indicator will be proofed in terms of the available word types. We defined an adjustable number of words and special word types which are significant for related units of text. The current word types are ADJA, ADJD, APPR, APPRART, CARD, PRELS and PROP [11], but more research has to be done to improve these rules based on the experiences during the project. If there are more elements in between, that do not comply with the given rules, the phrase will be closed. This kind of test will be done in both directions beginning from the central element of set B. The following noun chain will be decomposed the same way. Depending on the size of the gap between two noun chains, two phrases could be overlapping. In this case the two parts will be merged to one longer phrase. If not, the text in between will be marked as insignificant.

#### 2.4. Additional indicators for relations

Noun chains are based on structural and grammatical information. We are using further techniques which are crucially improving the results of the summarization process.

*Synonyms* Based on the nouns in the selected phrases we are collecting possible synonyms from the corpora of the "Leipzig Linguistic Services" [15]. The original full text is scanned for these synonyms. For each match a connection between the part with the synonym and the part with the original noun can be assumed. Moreover this link provides references for the corresponding importance.

*Phrases and linking words* Another way to filter important content is the usage of pre-defined phrases, that are common means of expression to emphasize the meaning of spoken or written material (e.g. "let's put this on the record"). Beyond that, there exist linking words, indicating connections between two text parts which can be used to merge phrases if found in between two important phrases. The problem here is to define an appropriate dictionary because the phrases and the vocabulary differ a lot depending on the situation and the text-style. The dictionary could be enhanced over a longer period through a self-learning mechanism. Dates and numbers can also be used to find potential significant content. Especially in meetings, significant parts like deadlines or amounts of money, usually contain numbers.

#### 2.5. User centered vocabulary

One focus of the project is to find a balance between the common view of significance and the user specific view in a particular context. Obviously, every particular user can have different opinions of what are the most important facts of a text. To allow a kind of individualization here, the user can define a set of documents which contains relevant vocabulary for the project. The material can be provided in all standard text formats, like PDF, Word or Email, and will be searched for special expressions which are not included in common dictionaries. We are assuming that these terms reflect user specific vocabulary (e.g. product or company names, proper nouns and abbreviations). Subsequently the system creates a custom dictionary which is used for customized corpora and to assign a higher ranking to the surrounding text parts.

### 3. Conclusion and future work

In this article we gave a short overview of our research project "Autoprot" and explained specific characteristics of spoken dialogues. We focused on the automatic summarization of meeting transcripts and described different approaches like noun chains, the use of synonyms and specific linking words as well as user centered dictionaries. In the future we will refine our rules to find noun chains, extend the use of phrase-level MALT-Parsing results to detect indicators and combine all previously mentioned techniques to improve the overall results.

### References

- [1] Ulrike Pado, Matthew Crocker and Frank Keller: A Probabilistic Model of Semantic Plausibility in Sentence Processing. *Cognitive Science*, 33(5):794-838, 2009
- [2] Matthew Crocker: Computational Psycholinguistics. In: Lappin, Clark and Fox (eds) *The Handbook of Computational Linguistics and Natural Language Processing*, Blackwell, UK, 2010.
- [3] Afra Alishahi and Suzanne Stevenson: Learning general properties of semantic roles from usage data: a computational model. *Language and Cognitive Processes*, 2010
- [4] Berg, Markus M., Isard, Amy and Moore, Johanna D.: An OpenCCG-Based Approach to Question Generation from Concepts. In: *Natural Language Processing and Information Systems*, pages 38-52, Springer Berlin Heidelberg, Lecture Notes in Computer Science 7934, 2013
- [5] Markus Berg, Antje Düsterhöft and Bernhard Thalheim: Towards Interrogative Types in Task-oriented Dialogue Systems. In: *17th International Conference on Applications of Natural Language Processing to Information Systems*, Springer, Groningen (The Netherlands), Lecture Notes in Computer Science 7337, 2012
- [6] Markus Berg, Antje Düsterhöft and Bernhard Thalheim: Query and Answer Forms for Sophisticated Database Interfaces. In: *22nd European Japanese Conference on Information Modelling and Knowledge Bases*, Prague (Czech Republic), 2012.
- [7] Markus Berg, Bernhard Thalheim and Antje Düsterhöft: Dialog Acts from the Processing Perspective in Task Oriented Dialog Systems. In: *Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2011)*, pages 176-177, Los Angeles (USA), 2011.
- [8] Gabriel Murray, Steve Renals, Jean Carletta: Extractive Summarization of Meeting Recordings, In *Proceedings, Interspeech'2005 - Eurospeech, 9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, September 4-8, 2005
- [9] Sven Hromadka: Keyword generation of full-text and semantic condensation of protocol documents, Masterthesis University of Wismar, 2013
- [10] Intellexer Summarizer, EffectiveSoft Inc., 2014
- [11] Heike Telljohann, Erhard W. Hinrichs: Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z), 2004
- [12] Francis, W. Nelson, and Henry Kucera. "Brown corpus manual." Brown University Department of Linguistics, 1979
- [13] Copernic Summarizer, Copernic Inc., 2014
- [14] Brants, Sabine, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit: TIGER: Linguistic Interpretation of a German Corpus. *Journal of Language and Computation*, 2004 (2), 597-620
- [15] Marco Büchler, Gerhard Heyer: Leipzig Linguistic Services - A 4 Years Summary of Providing Linguistic Web Services, In: *Proceeding of TMS 2009, Leipzig, Germany, March 2009*
- [16] Yatsko, V. A., and T. N. Vishnyakov: A method for evaluating modern systems of automatic text summarization. *Automatic Documentation and Mathematical Linguistics* 41.3, pages 93-103, 2007

# Intercultural Collaboration in Virtual Environment

Tatjana Welzer<sup>a,1</sup>, Hannu Jaakkola<sup>b</sup>, Marko Hölbl<sup>a</sup>, Marjan Družovec<sup>a</sup>, Anthony. E. Ward<sup>c</sup>

<sup>a</sup>*University of Maribor  
Faculty of Electrical Engineering and Computer Science  
Maribor, Slovenia*

<sup>b</sup>*Tampere University of Technology, Pori  
Pori, Finland*

<sup>c</sup>*University of York  
York, UK*

**Abstract.** For successful collaboration are nowadays mostly more important the global communication systems, which are enabled by information and communication technology as partners in the process. It seems that partners do not need to know each other personally, but it is enough that they have a good technical support in computerized collaborative tools and software. This also means that collaboration (cooperation) is irrespective of the geographical location and can be spread all around the world. Establishing a virtual environment means establishing a surrounding in which partners have to work and communicate. From technical point of view such an environment seems to be quite easy to organise and manage. Most probably difficulties can be expected in the collaboration of people, even that they should cooperate easy on the base of common business culture. In spite the same business culture, participants can belong to different language and culture groups what can have also an important influence on collaboration of all involved parties. While the language differences can be under the control by using English as lingua franca, we do not have a Culture as culture franca. This leads attention to intercultural collaboration in the virtual environment. Education in cultural studies, cultural awareness and appropriate tools will be needed. In the paper we will point out all important components for international collaboration in virtual environment from definitions about intercultural and collaboration up to methodologies, that supporting work in virtual environment and virtual centres itself.

**Keywords.** Interculture, language, collaboration, virtual environment

## Introduction

It seems that successful collaboration is nowadays mostly more dependent on the global communication systems as on partners. That means that partners do not need to know each other personally, but it is enough that they have a good technical support in information and communication technology, like computerized collaborative tools and supporting software. This also means that collaboration and cooperation is possible

---

<sup>1</sup> Corresponding Author: University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova ulica 17, 2000 Maribor, Slovenia; E-mail: tatjana.welzer@um.si

irrespective of the geographical origin of the partner and can be spread all around the world. The collaboration presented in the paper will be concentrated in observing international participants in the virtual environment with defined roles and different background in expertise (expert topic). As the case study we are observing VCE a virtual education centre for the development of entrepreneurial skills and competences which is called Virtual Centre for Entrepreneurship (VCE) [1], [17]. The centre is preparing intensive vocational courses (e-learning modules), mostly for employed learners. [1]. Providers of e-learning modules are partners of the VCE and are coming from higher education institutions all over Europe. Possible partners would be also other educational organisations and companies interested in dissemination and sharing of knowledge [14].

This also means that providers, learners and others are coming from different language and cultural environments, what have as we expected an important influence on collaboration of all involved parties. Also different experiences of partners have to be taken into account.

Further, we will present a structure of the VCE and stress the importance of diversity of languages as well as cultural awareness. At last but not least we will present experiences of partners collaborating in the VCE activities according to the roles that they have in the system.

## **1. VCE – presentation, structure, key actors**

The VCE was established to provide training in the field of Electrical and Information Engineering, but there are no limits for covering any other topics or area if providers and users of e-modules exist. The facilities of the centre are open to learners at all levels of their education and to individuals of all ages who wish to engage. [1]. Between activities supported by the VCE, that we would like to point out, are the following [4], [17]:

- E-learning system that individuals can register and select a module that would like to work on it. Whereby learners can also develop their language skills in a foreign language as part of their learning.
- Teaching resources (VCE e-modules) available for partners for freely use for their own teaching purposes.
- A reference repository of relevant research publications in fields like Electrical and Information Engineering, Pedagogy, Assessment and others.

Each VCE e-module has clear defined aims by means of Learning 2.0, presented through professional content, communication, collaboration and differentiation (Figure 1) [12]. Possible obstacles exist because of different languages, multicultural participation and using of a virtual environment (Figure 1) [12]. Especially important are multilingualism and multiculturalism as obstacles. Participants collaborating in virtual environment have to be aware of them.

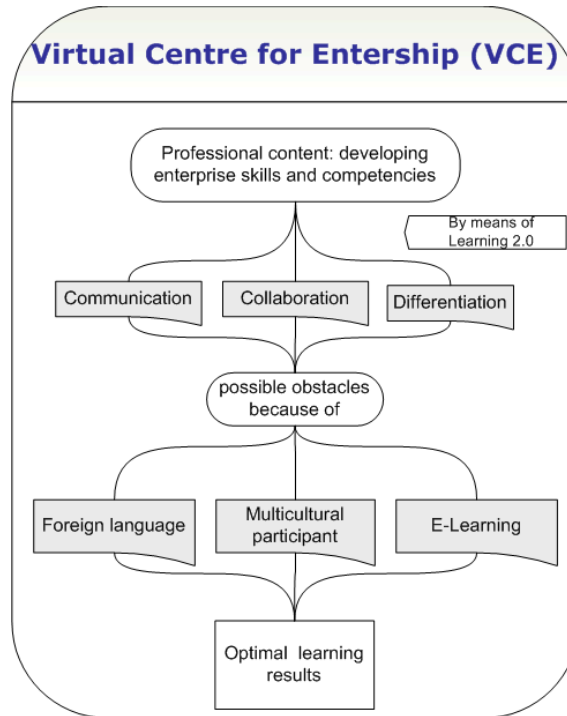


Figure 1: VCE e-module concepts [12]

Because of participants from different cultures and language groups, we expect usage of different languages, what will give the VCE possibility for collaboration all around the world using VCE 2, supporting different language groups and providing collaboration in additional languages beside English. The VCE welcome page is in the sense of those plans available in 5 different languages: English as the basic language, French, Spanish, Turkish and Slovene. Figure 2 [16], [18] is presenting just three of them – English, Spanish and Slovene.



Figure 2: VCE welcome page in 3 languages [16],[18]

Participants in the VCE have different roles. Key actors are learners, tutors, teachers, moderators and we have also some others. In some cases, a participant can

appear as the same person in different roles - if the case is extreme, also in all possible. So far we want to point out the following roles [4], [17]:

- Learner – participant involved with the VCE, with the intention of undertaking learning.
- Mentor/tutor/teacher – participant approved to mentor/tutor/teacher learn in one or more VCE e-modules.
- Mentor moderator – participant approved to view the credentials of a prospective mentor and approve the individual to be a mentor/tutor/teacher.
- Developer/producer – participant who is developing/producing the VCE e-modules.
- Resource moderator/editor – participant approved to view submitted learning VCE e-modules and approves them as usable.
- Visitor – participant who wishes to access the VCE and has to registered also only to visit the VCE and not take the part in learning.
- Administrator – participant who is administrating the VCE and the VCE environment (Moodle).

## **2. Concepts of Languages and Cultures in Intercultural Collaboration**

Language is extremely important for teaching and learning processes, even more if we are using virtual environment. Misunderstandings cause serious mistakes if words are imprecise [12]. In VCE, most participants will use e-modules in for them a most probably foreign language. Exceptions are English native speakers while VCE e-modules are in this version only in English language. But independently of e-module in VCE all learners could be faced with the language problem, while tutors and other VCE staff can speak different language as learner and already mentioned misunderstandings can happen. Also ideas can get lost and learner's role in collaborative learning activities can be missed or minimised. Due to language barriers, learner's organisation and timing of study are influenced. Learner probably need more time for learning and understanding learning material as well as answering or preparing some other materials demands in the course. We have to deal with a lower level of communication and participation in different forms of discussions. On the other side participants have a great opportunity to improve the language skills and extend expert and non-expert vocabulary.

According to R.D.Lewis is language poor communication tool unless each word or phrase is seen in its original cultural context [7]. This means that we have to take care also about the cultural awareness. In literature, we are confronted with many definitions of culture. Hofstede defines culture as a collective phenomenon, because it is shared with people who live or lived within the same social environment. According to these definitions, culture consists of unwritten rules of social game and is the collective programming of the mind that distinguishes the member of one group or category of people from others [5], [6]. For daily use and especially for the collaboration in the virtual environment is more than definition of culture important to clarify cultural awareness. No or poor cultural awareness means poor understanding of the intercultural dialogue, which probably can lead to blunders and damaging consequences. Especially sensitive are expert areas like business, management and advertising [8], where cultural awareness seems to be of key importance for success and respect of rules based on

culture or/and religion. However, engineering and many other areas are also not immune against it [9]. If cultural awareness is the key of business, it should not be missing in other areas, including engineering and medicine. For that reason, knowledge of cultural awareness should not be missed in any of e-module in VCE.

From theoretical point of view, cultural awareness is the foundation of communication, and it involves the ability of observing our cultural values, beliefs and perceptions from the outside [2]. It becomes important in communication with people from other cultures, and we have to understand that people from different cultural environments can see, interpret and evaluate things in different ways [12], [11], [16]. Becoming aware of culture is a difficult task since culture is not conscious of us – it is like water to fish – we live and breathe through it [10]. To avoid these cultural obstacles and follow the presented definitions, teachers/tutor/mentors as professional experts with the knowledge of culture diversity have to introduce into VCE e-modules examples and discussions that are covering different cultural points of view offering different way of communication, collaboration and expression supporting cultural diversity.

### **3. Collected experiences and Conclusions**

We would like to stress experiences that we have collected from first learners and tutors/teachers as well as developer/producer – the whole system is still in the prototype phase. Roles except administrator are not fully functional, because the procedures for being approved for the key actor are not finalised yet. To the same time administrator seems not to be involved in language and culture obstacles, while activities will be done on one place in one language and culture. But with possible distribution of VCE (VCE 2) [15] this can turn into the similar problem, that can leads to even more complex obstacles as those already introduced. Nevertheless the problem could appear also between administrators, tutors/teachers/mentors and developers/producers but not from the point of collaboration in the VCE, but the content of e-modules.

The most often comments from learners are the following: they prefer the approach and collaboration in the virtual environment, they prefer the amount of learning material (limited to the small number of ECTS, they have some problems with e-learning tool, they expect more culture oriented examples including business cultures, more detailed explanations, tutor's availability and life contact. language itself seems not to be problem at least for most of learners already registered in the VCE.

Tutors/teachers/mentors as well as developers/producers are faced with other problems: teaching face to face is easier, while development of VCE e-modules is time consuming. Tutors/teachers/mentors have to be trained how to mentor and also how to teach, even more intensive training is needed for developer/producer while preparing an e-module (lecture) is much more pretending as in the case of face to face teaching and at last but not least quality assessment of courses is needed. No real language or culture problems appear, but more probably they are not aware of them because of own position – learners will probably keep for them those kinds of problems.

If we summarise, we tried in the paper to emphasise the importance of collaboration in systems which are active in multicultural and multilingual environments and which can easily assume international character, just as VEC does. In such e-environments, we have to be aware of presence of culture and appearance of

foreign languages in the dialogue between teachers/tutors/mentors and learners, teachers/tutors/mentors and developers as well as among teachers/tutors/mentors, developers and learners itself.

To sum up, cultural awareness in general means being open to ideas of changing cultural attitudes or being sensitive to the difference between how we would like to be perceived by others and how we are actually perceived by others. Cultural awareness recognizes that we are shaped by our cultural background, which influences how we interpret the world around us, perceive ourselves and how we relate to other people [8], [9].

Also problems with understanding different languages can appear. The collection of available comments did not point out that either learners, either teachers/tutors/mentors or developer would have any significant problems with foreign languages. Of course opinions have been collected on the relation learner-teacher/tutor/mentor and teacher/tutor/mentor – developer but not between teachers/tutors/mentors, developers and learners. At last but not least more information have to be collected also for cases that one participant can take different roles – will he/she change his/her opinion on some problems according to the role? This could probably change results, which are because of a small number of available answers not really relevant, but are good background for further development of collaboration in the VCE.

## References

- [1] H. Yahoui. ELLEIEC – Enhancing Lifelong Learning for the Electrical and Information Engineering Community, Project Report, 2010
- [2] S.C Schneider, J-L. Barsoux, Managing Across Cultures, Prentice Hall, Harlow, 2003.
- [3] T. Welzer, M. Družovec, P. Cafnik, M. Zorič Venuti, H. Jaakkola. Awareness of Culture in e-learning. In: ITHET 2010, IEEE, 2010, pp. 312-315.
- [4] C. Perra, H. Yahoui, T. Ward. An Entrepreneurship Center in the lifelong learning spirit. In: ITHET 2010, IEEE, 2010, pp. 215-218.
- [5] G. Hofstede. Culture's Consequences, Comparing Values, Behaviors, Institutions and Organizations Across Nations. Sage Publications, Thousand Oaks, 2001.
- [6] G. Hofstede, G.J. Hofstede. Cultures and Organizations: Software of the Mind: Intercultural Cooperation and its Importance for Survival. McGraw-Hill, New York, 2004.
- [7] R.D. Lewis. When Cultures Collide, Managing Successfully Across Cultures. Nicholas Brealey Publishing, London, 2007.
- [8] B. Farsides. Cultural Awareness and Common Understanding: The Key to Informed Consent? <http://www.tbethics.org/conf18.htm>, last visit March 7, 2014.
- [9] T. Welzer, M. Bonačić, M. Zorič Venuti. Cultural awareness in information society. In: Proceedings of 20th EAEEIA annual conference Innovation in education for electrical and information engineering, 2009, IEEE, 2009, 4 f.
- [10] T. Welzer. Cultural and security issues in knowledge management. In: Proceedings of Znalosti 2009, (Edícia zbornikov Informatiky a informačných technológií), 2009, pp. 25-29.
- [11] S.Quappe, G. Cantatore. What is Cultural Awareness anyway? How do I build it?. Cultuosity.com.
- [12] M. Zorič Venuti, T. Welzer, A.E.Ward. Key Issues Concerning the Simultaneous Development of Technical and Linguistic Components in a single E-Module In: DeSE 2010, IEEE, 2010.
- [13] T. Welzer, M. Zorič Venuti, A.E.Ward, H.Yahoui. Implementation of an E-learning Module in Virtual Centre for Entrepreneurship: The development of cultural awareness in students In: DeSE 2010, IEEE, 2010.
- [14] P. Linna, E. Karttunen, H. Jaakkola. Software Engineering Companies' Multicultural Education. In: MIPRO 2011, IEEE pp. 177-182.
- [15] H. Jaakkola, P. Linna, J. Henno. (Social) networking is coming - are we ready? In: MIPRO 2011, IEEE pp.170-176.



- [16] T. Welzer, M. Zorič Venuti, A.E.Ward. Collaboration in the virtual center for entrepreneurs. In: Proceedings of the 13th International Multiconference Information Society - IS 2010, volume A, (Informacijska družba, ISSN 1581-9973). Ljubljana: Institut Jožef Stefan, 2010, pp. 255-258
- [17] C. Perra, H. Yahoui, T. Ward. A Virtual Center for Entrepreneurship. In: Elektronika ir elektrotehnika, ISSN 1392-1215. [Print ed.], 2010, nr. 6 pp. 113-116.
- [18] T. Welzer, M. Zorič Venuti, A.E.Ward, M. Hölbl, M. Družovec. Virtual education centre for the development of expert skills and competencies. International journal of advanced corporate learning. 2011, vol. 4, no. 4, pp. 51-54

## Author Index

AlBdaiwi, Bader	1
Alenizi, Hanieah	28
AlMazyad, Weaam	20
Alnajran, Nouf	36
AlOthman, Fahdah	20
AlOthman, Fahdah	28
Alsaggyer, Hadeel	28
Alshareed, Ahlam	28
Alswilmi, Mashail	36
Aramvith, Supavadee	143
Ardalani, Peyman	494
Atzeni, Paolo	44
Berg, Markus	516
Britell, Scott	44
Brumen, Boštjan	63, 99
Burita, Ladislav	71
Černežel, Aleš	63, 99
Chauksuvanit, Teeranoot	143
Chen, Xing	79
Čihalová, Martina	108
Czech, Gerald	473
Dahanayake, Ajantha	20, 28, 36
Delcambre, Lois	44
Družovec, Marjan	521
Düsterhöft, Antje	230, 516
Duži, Marie	347
Eigenstetter, Christoph	230, 516
Endrjukaite, Tatiana	123
Fettke, Peter	494
Fukamachi, Ken-ichi	158
Gerlich, Jakub	347
Gogolla, Martin	428
Hashimoto, Takako	143
Hayashi, Yasuhiro	158
Heimbürger, Anneli	166, 248, 508
Henno, Jaak	178
Hölbl, Marko	521
Itabashi, Yoshiko	190
Jaakkola, Hannu	210, 521
Jakunshin, Jevgenij	230
Kärkkäinen, Tommi	248, 508
Kaschek, Roland	235
Khanom, Sukanya	248
Kiyoki, Yasushi	123, 190, 380, 413, 448
Komatsugawa, Hiroshi	158
Košinár, Michael Alexander	261, 464
Kramer, Frank	281
Leber, Diethard	473
Liehr, Clemens	473
Locuratolo, Elvira	300, 320
Lomov, Pavel	339
Loos, Peter	494
Menšík, Marek	347
Mizuoka, Atsushi	399
Nakanishi, Takafumi	364
Nguyen, Diep Thi-Ngoc	380
Noack, René	1
Noro, Tomoya	399
Ondryhal, Vojtech	71
Palomäki, Jari	320, 407
Patrisia, Soffi	413
Pramadihanto, Dadet	413
Rozman, Ivan	63
Sasaki, Shiori	190, 413
Sedlmeier, Matthias	428
Sesulihatien, Wahyu T	413, 448
Shiohara, Keiichi	79
Shirota, Yukari	143
Shishaev, Maxim	339
Štolfa, Jakub	261, 464
Štolfa, Svätopluk	261, 464
Štrba, Radoslav	464
Thaler, Tom	494
Thalheim, Bernhard	1, 210, 281, 473
Tokuda, Takehiro	399
Tropmann-Frick, Marina	473
Yoshida, Naofumi	486
Walter, Jürgen	494
Ward, Anthony. E.	521
Wartiainen, Pekka	508
Weber, Nils	516
Weltzer, Tatjana	521