Janis Grundspenkis
Marite Kirikova
Yannis Manolopoulos
Leonids Novickis (Eds.)

LNCS 5968

Advances in Databases and Information Systems

Associated Workshops and Doctoral Consortium
of the 13th East European Conference, ADBIS 2009
Riga, Latvia, September 2009
Revised Selected Papers

Lecture Notes in Computer Science 5968

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Janis Grundspenkis Marite Kirikova
Yannis Manolopoulos Leonids Novickis (Eds.)

Advances in Databases and Information Systems

Associated Workshops and Doctoral Consortium
of the 13th East European Conference, ADBIS 2009
Riga, Latvia, September 7-10, 2009
Revised Selected Papers

Springer

Volume Editors

Janis Grundspenkis
Riga Technical University, Dept. of Systems Theory and Design
Meza str 1/4, Riga, Latvia, LV 1048
E-mail: janis.grundspenkis@rtu.lv

Marite Kirikova
Riga Technical University, Institute of Applied Computer Systems
Meza str 1/4, Riga, Latvia, LV 1048
E-mail: marite.kirikova@rtu.lv

Yannis Manolopoulos
Aristotle University, Dept. of Informatics
Thessaloniki, 54124, Greece
E-mail: manolopo@csd.auth.gr

Leonids Novickis
Riga Technical University, Division of Applied Computer Systems Software
Meza str 1/4, Riga, Latvia, LV 1048
E-mail: leonids.novickis@rtu.lv

Library of Congress Control Number: 2010922319

CR Subject Classification (1998): H.2, H.3, H.4, H.5, D.2, J.1

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-642-12081-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-12081-7 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Editors' Preface for Associated Workshops Section

Five workshops (INTEL-EDU, MEDWa, XSchemas, TR4Web, MDA) were organized in conjunction with the 13th East-European Conference on Advances in Databases and Information Systems (ADBIS 2009) held September 7-10, 2009, in Riga, Latvia:

- INTEL-EDU - Intelligent Educational Systems and Technology-Enhanced Learning
- MEDWa - Managing Evolution of Data Warehouses
- XSchemas - Schema Languages for XML
- TR4Web - Trust, Risk, Reputation and Recommendation on the Web
- MDA - Model-Driven Architecture: Foundations, Practices and Implications

In a two-step reviewing process the workshop Program Committees accepted 30 papers to be published in the ADBIS 2009 Associated Workshops and Doctoral Consortium conference proceedings.

The ADBIS 2009 workshops aimed to create conditions for experienced and young researchers to share their knowledge and to promote collaboration between European research communities (especially from Central and East Europe).

The accepted papers of the **INTEL-EDU** workshop cover a wide spectrum of topics on innovative adaptive and intelligent systems for learning, advanced cognitive tutors, virtual reality in the training process, managing and organizing e-Learning systems, agent-based simulation, psychophysiological model-based adaptive e-learning systems, concept map-based intelligent knowledge assessment systems, quality of study programs.

The aim of the **MEDWa** workshop was to gather researchers who concentrate their works on handling various aspects of a data warehouse evolution and to provide a forum for discussing their achievements and open issues. The workshop also included keynote talk, "Allegro's Way from XLS-Based Controlling to a Modern BI Environment," by Christian Maor.

The **X-Schemas** workshop was focused on bringing together researchers that are interested in sharing new ideas related to XML schema languages. The presence of a schema is crucial to data exchange, and can facilitate the automation and optimization of integration, processing, search and translation of XML data.

The **MDA** Workshop was aimed at theoretical and practical aspects of OMG's Model-Driven Architecture and Model-Driven Development as well.

The **TR4Web** workshop was used to present results in the fields of risk, reputation, recommendation and trust on the Web environments. The workshop included keynote talks (joint for MEDWa and TR4Web).

We would like to thank the authors, who submitted papers, and the program committees members, who did a great job of paper reviewing. We acknowledge the ADBIS 2009 Organizing and Steering Committees for their cooperation in organizing the workshop sessions. We hope that you find the results presented here interesting and useful.

September 2009

Janis Grundspenkis
Yannis Manolopoulos
Irena Mlynkova
Mikolaj Morzy
Martin Necasky
Oksana Nikiforova
Leonids Novickis
Janis Osis
Robert Wrembel

Conference Organization

Executive Committee

General Chair	Janis Grundspenkis (Riga Technical University, Latvia)
Program Chair	Tadeusz Morzy (Poznan University of Technology, Poland)
	Gottfried Vossen (University of Münster, Germany)

Program Committee

Paolo Atzeni	Italy
Guntis Barzdins	Latvia
Andreas Behrend	Germany
Andras Benczur	Hungary
Maria Bielikova	Slovakia
Bostjan Brumen	Slovenia
Alina Campan	USA
Albertas Caplinskas	Lithuania
Sharma Chakravarthy	USA
Alfredo Cuzzocrea	Italy
Alin Deutsch	USA
Johann Eder	Austria
Janis Eiduks	Latvia
Johann Gamper	Italy
Jarek Gryz	Poland
Hele-Mai Haav	Estonia
Theo Haerder	Germany
Mirjana Ivanovic	Serbia
Hannu Jaakola	Finland
Manfred A. Jeusfeld	The Netherlands
Leonid Kalinichenko	Russia
Ahto Kalja	Estonia
Audris Kalnins	Latvia
Marite Kirikova	Latvia
Margita Kon-Popovska	FYRO Macedonia
Sergei Kuznetsov	Russia
Mehmed M. Kantardzic	USA
Maurice van Keulen	The Netherlands
Jens Lechtenboerger	Germany

Nikos Mamoulis	China
Yannis Manolopoulos	Greece
Rainer Manthey	Germany
Joris Mihaeli	Israel
Pavol Navrat	Slovakia
Igor Nekrestyanov	Russia
Mykola Nikitchenko	Ukraine
Kjetil Norvag	Norway
Boris Novikov	Russia
Gultekin Ozsoyoglu	USA
Tamer žsu	Canada
Evi Pitoura	Greece
Jaroslav Pokorny	Czech Republic
Boris Rachev	Bulgaria
Peter Revesz	USA
Tore Risch	Sweden
Stefano Rizzi	Italy
Peter Scheuermann	USA
Timos Sellis	Greece
Vaclav Snasel	Czech Republic
Eva Soderstrom	Sweden
Nicolas Spyratos	France
Janis Stirna	Sweden
Val Tannen	USA
Bernhard Thalheim	Germany
Juan Trujillo	Spain
Olegas Vasilecas	Lithuania
Michael Vassilakopoulos	Greece
Krishnamurthy Vidyasankar	Canada
Gerhard Weikum	Germany
Marek Wojciechowski	Poland
Limsoon Wong	Singapore
Shuigeng Zhou	China

Workshops Co-chairs

Yannis Manolopoulos	Greece
Leonids Novickis	Latvia

ADBIS Steering Committee

Leonid Kalinichenko	Russian Academy of Science, Russia (Chair)
Andras Benczur	Hungary
Albertas Caplinskas	Lithuania
Johann Eder	Austria
Marite Kirikova	Latvia

Hele-Mai Haav	Estonia
Mirjana Ivanovic	Serbia
Mikhail Kogalovsky	Russia
Yannis Manolopoulos	Greece
Rainer Manthey	Germany
Manuk Manukyan	Armenia
Joris Mihaeli	Israel
Tadeusz Morzy	Poland
Pavol Navrat	Slovakia
Boris Novikov	Russia
Mykola Nikitchenko	Ukraine
Jaroslav Pokorny	Czech Republic
Boris Rachev	Bulgaria
Bernhard Thalheim	Germany
Tatjana Welzer	Slovenia
Viacheslav Wolfengagen	Russia
Ester Zumpano	Italy

Organizing Committee Chairman

Agris Nikitenko	Riga Technical University, Latvia

Organizing Committee Members

Janis Eiduks	Riga Technical University, Latvia
Larisa Survilo	Riga Technical University, Latvia
Dace Apshvalka	Riga Technical University, Latvia
Marite Kirikova	Riga Technical University, Latvia
Uldis Sukovskis	Riga Technical University, Latvia
Juris Borzovs	Latvian IT Cluster, Latvia
Lilita Sparane	Latvian IT Cluster, Latvia

ADBIS 2009 Doctoral Consortium Co-chairs

Zohra Bellahsène	University of Montpellier II, France
Atis Kapenieks	IEEE, Latvia
Marite Kirikova	Riga Technical University, Latvia

Editors' Preface for Doctoral Consortium Section

The Doctoral Consortium of the 13th East-European Conference on Advances in Databases and Information Systems (ADBIS 2009) held September 7-10, 2009, in Riga, Latvia, was organized with the purpose to facilitate high-quality research in the field of database and information systems. The Doctoral Consortium attracted 18 submissions from 4 countries, namely Latvia, Lithuania, Russia, and Poland. In a two-step reviewing process the international Program Committee of 22 members from 17 countries accepted 7 papers for the ADBIS 2009 Associated Workshops and Doctoral Consortium conference proceedings. The accepted papers cover topics on software development, business process modeling, conceptual modeling, XML schemes, clustering and structuring of data, and location-based information storage.

According to the above-mentioned purpose of the Doctoral Consortium, the Program Committee tried to help young researchers to qualify for publication in the prestigious conference proceedings, which is still a problem in Eastern European countries. All doctoral students received valuable reviews on how to improve their contributions. The Doctoral Consortium was organized as parallel sessions in the ADBIS 2009 conference in order to create conditions for experienced researchers to communicate their knowledge and experience to the young researchers participating in the Doctoral Consortium. Additionally, a discussant from the Doctoral Consortium Program Committee was assigned to each presentation to facilitate focused and deep discussions on considered research topics.

We would like to express our thanks to all the people and organizations who contributed to the success of the ADBIS 2009 Doctoral Consortium. We thank the doctoral students, who submitted papers to the consortium and diligently worked on improving them up to a high level of quality, the Program Committee members who worked hard in reviewing the Doctoral Consortium papers and suggesting ways for improvement; and we thank those Program Committee members who agreed to become discussants at the Doctoral Consortium sessions. We acknowledge the ADBIS 2009 Steering Committee for encouraging and helping us to organize the Doctoral Consortium. We are grateful to the VLDB Society that generously sponsored the Doctoral Consortium in these financially stressed times. We also acknowledge the Riga Technical University for its continuous support and the Latvian University and IEEE Latvia for assistance in organizing the ADBIS 2009 Doctoral Consortium.

September 2009

Zohra Bellahsène
Atis Kapenieks
Marite Kirikova

ADBIS 2009 Doctoral Consortium Organization

Doctoral Consortium Co-chairs

Zohra Bellahsène	University of Montpellier II, France
Atis Kapenieks	Riga Technical University and IEEE Latvia
Marite Kirikova	Riga Technical University, Latvia

Programme Committee

Andris Ambainis	Latvian University, Latvia
Guntis Barzdins	Latvian University, Latvia
Albertas Caplinskas	Institute of Mathematics and Informatics, Lithuania
Janis Grabis	Riga Technical University, Latvia
Janis Grundspenkis	Riga Technical University, Latvia
Hele-Mai Haav	Tallin Technical University, Estonia
Ela Hunt	Strathclyde University, UK
Leonid Kalinichenko	Russian Academy of Science, Russia
Yannis Manolopoulos	Aristotle University of Thessaloniki, Greece
Tadeusz Morzy	Poznan University of Technology, Poland
Agris Nikitenko	Riga Technical University, Latvia
Jaroslav Pokorny	Charles University, Czech Republic
Boris Rachev	Technical University of Varna, Bulgaria
Mark Roantree	Dublin City University, Ireland
Leo Selavo	Latvian University, Latvia
Pnina Soffer	University of Haifa, Israel
Ernest Teniente	Technical University of Catalonia, Spain
Farouk Toumani	Blaise Pascal University, France
Benkt Wangler	University of Skovde, Sweden
Tatjana Welzer	University of Maribor, Slovenia
Wita Wojtkowski	Boise State University, USA
Ester Zumpano	University of Calabria, Italy

Organizing Committee

Ilze Birzniece	Riga Technical University, Latvia
Girts Karnitis	Latvian University, Latvia
Ludmila Penicina	Riga Technical University, Latvi

Table of Contents

Doctoral Consortium Section

Virtual Reality Platforms for Education and Training in Industry

Eberhard Blümel and Tina Haase

Fraunhofer Institute for Factory and Automation,
Sandtorstr. 22, 39106 Magdeburg, Germany
Eberhard.Bluemel@iff.fraunhofer.de, Tina.Haase@iff.fraunhofer.de
http://www.vdtc.de

Abstract. Developing, testing and operating complex machinery and repairing it under time pressure if it breaks down are some of the new skills, professionals in many occupations have to learn as quickly as possible. Actions on machinery and plants are trained in individual lessons on an immersive virtual model. This paper introduces the methodology behind the technical solution and presents experiences acquired during its implementation with a virtual learning platform for operators and maintenance staff as examples.

1 The Fraunhofer IFF Learning Platform

1.1 General Concept

The training concept upon which the Fraunhofer IFF learning platform is based allows customizing training to match future operators' levels of knowledge by variably configuring the level of difficulty on the basis of the level of interaction. A process to be learned can be explored initially in a model solution. First, learners are introduced to the entire process as well as its interactions with other subsystems. After that, learners are requested to execute the process, independently completing a task by interacting with the virtual system. The learning platform's system initially supports them by continuing to issue concrete instructions on completing the individual process steps.

Afterward, learners ought to be able to execute the assigned tasks without the system's help. The learning platform logs all of a learner's activities as well as the time and number of attempts required to complete a task. This information is referenced later to assess training performance.

The Fraunhofer IFF has developed a software platform that enables training and interaction with realistic virtual products, machinery and plants on the basis of 3D immersive virtual environments. A new technology has been produced that enables trainers to conduct and trainees to partake of theoretical and practical training on complex models individually and technical training in teams in distributed environments without having to revert to real objects. The recognition value of the visualization, the realism of the simulation of a product or plant's

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 1–7, 2010.

behaviour and options for realistic user interaction are essential. Numerous cross-industry solutions for virtual interactive learning platforms for different training and educational objectives ([1], [2], [5], [7]) have been developed.

A Tutoring System for Web-Based and Case-Oriented Training in Medicine is presented in [4]. It includes a training system that is based on three different models:

- Tutoring Process Model
- The Case Knowledge Model
- The Medical Knowledge Model.

Those models are chosen and adapted to the domain of medicine. Although it is very domain specific, it shows certain parallels to the approach presented in this paper, e.g. a knowledge base that allows users to receive information on demand and a guided learning tour that supports the learning process step by step.

The IFF training solutions are conceptually based on the precise objectives of the VR learning environment for the case of application and the methodical analysis of the skills to be learned (perception, orientation, skills, communication, etc.), users' personalities and learning styles and the structure of the course to teach the training objectives. The underlying concept is based on the methodical analysis of various assembly and maintenance tasks and classified features of applicable training environments based on virtual technologies that serve as the starting point for modeling a learning environment. This concept requires a supporting VR system architecture.

1.2 The Layered Architecture

A technical system must be modeled realistically to obtain realistic training conditions. Thus, a model should react and also respond to user actions just as the real equipment. Users must be enabled to perform every relevant action they would in the real world in the simulated environment (cf. [3]). A great deal more information has to be modeled in addition to the objects' geometry, e.g. the hierarchy of objects and possible parenting relationships, constraints on movement, causalities, properties, actions and dynamic behavior. Furthermore, components are needed to enable trainees to evaluate their actions themselves and to facilitate communication between trainees and instructors.
The information needed to model a training environment can be divided into three levels (see Figure 1).

- Geometry level: This level includes every type of node (geometry, animation, trigger, level-of-detail switches, etc.) common to the scenario structure of most existing VR systems. These entities provide the formal basis for implementing a scenario in a runtime system. Suitable converters import the information on this level from other systems such as CAD applications. Engineers, instructors and educators normally do not have to know details on these levels.

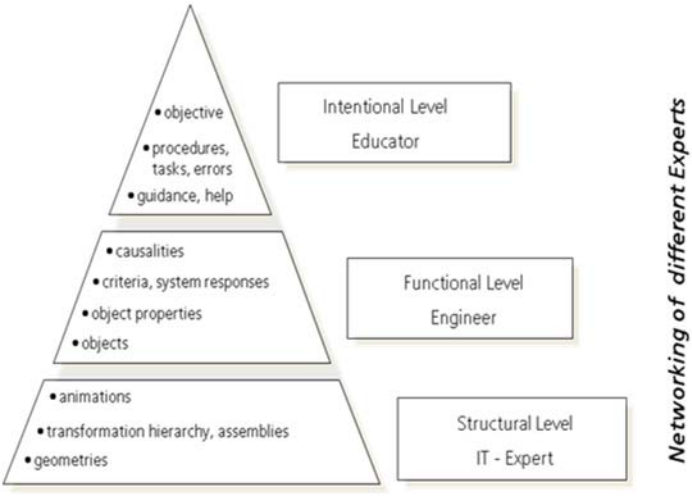


Fig. 1. The levels of the VR scenario concept

– Object level: This level is the domain of design engineers and contains the technological know-how specific to a system.

  The object level contains all information specific to a product already defined in the design process. It also includes characteristics determined by natural constraints, e.g. gravity, collision detection/prevention, etc.
– Instructional level: This level is the domain of educators or instructors. Objects defined on the object level can be utilized here to form training tasks. Training tasks can be used to construct lessons. One or more lessons may be necessary to attain a certain training objective.

All three of the aforementioned levels are interdependent and each level requires specialists from different fields.

Suitable tools that fulfill both, the technical and functional and the pedagogical and didactic aspects of the content, are needed to implement the substantial technical know-how required for existing technical options, cost effectively as well. To attain the high flexibility required by the applications developed, the components were divided in:

– Authoring system: The authoring system provides trainers support when they create training scenarios. The authoring system is intended to provide trainers a tool that requires a minimum of knowledge of computers. Unlike most training systems developed by computer experts, this application is intended to grant experts in the field of training diverse options for creativity.

- Scenario data: Work with the authoring system produces a training scenario saved as a scenario file containing the specific data for a concrete training task.
- Runtime system: The runtime system is essential to conducting training. It is independent of the concrete training scenario and can be equally applied to every scenario created with the authoring system. One runtime system instance is required for every trainer and trainee.

Building upon this work, the Fraunhofer IFF developed tools that simplify the creation of training scenarios from the conversion of design data up through the creation of complex causal chains to define training objectives and generate training tasks even without knowledge of complex programming.

2 Best Practice Examples: Virtual Interactive Training for RWE AG

In the course of refining its internal training concept, RWE's Technik Center Primärtechnik (TCP) decided to collaborate with the experts from the Fraunhofer IFF Virtual Development and Training Centre VDTC.

The specific constraints of ongoing technical operation, which make training in real situations quite difficult, were the reason for doing so. One far-reaching problem is the relative impossibility of using sensitive equipment in operation, in this case transformers, for training for reasons of safety and because they are integrated in national or international power grid structures. Moreover, the pertinent safety regulations must always be strictly observed whenever inspection, servicing, maintenance and improvement work is being performed. This also complicates training considerably. In addition, it is impossible to really observe functional processes inside equipment. Therefore, specialists need a high degree of technical knowledge and the ability to think abstractly to understand them.

In the end, the stations' decades-long service life necessitates developing the know-how of the technical specialists for the company and making it useful for future generations. This must be done as simply, vividly and standardized as possible.

1. Preparation of a Transformer for Rail Transport
   The TCP uses a VR scenario of an extra high voltage grade power transformer of up to 200 MVA in the 220 kV capacity class. It shows interactive animations to train operators how to unplug it from the grid and disconnect the important elements (cf. [1]). These operations prepare the main transformer body for transportation and future use. The objective is for trainees to learn, train, comprehend and internalize the work procedures (see Figure 2).

   Given the high risks connected with the procedure, it is essential to demonstrate the correct sequential order of work incorporating safety regulations that guarantee safe conditions (cf. [1]). VR immerses trainees in

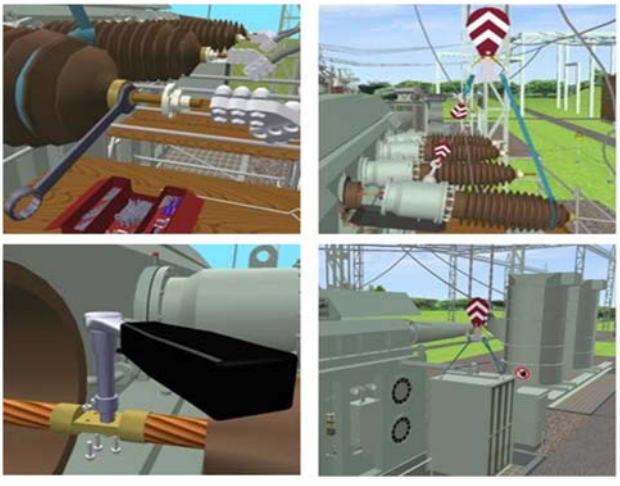


Fig. 2. Steps of the work procedure

realistic simulated environments. This can provide work experience without any risk of accidents or maloperation. Learning and interacting with VR is more efficient than traditional methods.

2. Replacing a Buchholz Relay
   Transformers are critically important equipment and their condition affects operations. A Buchholz relay is an important device that protects and monitors liquid cooled transformers and compensation reactors. It is easy to operate, highly reliable, maintenance free and long lived.

   A relay is installed in the connecting pipe between the transformer tank and the conservator tank (see Figure 3). In normal operation, it is filled completely with insulating liquid. The float is buoyed to its highest position. In reaction to malfunctions inside a transformer, a Buchholz relay collects the

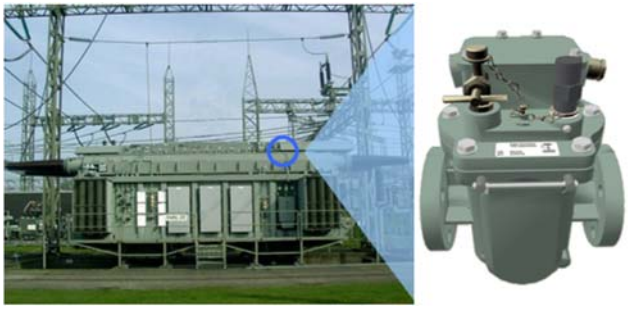


Fig. 3. Buchholz relay in a transformer

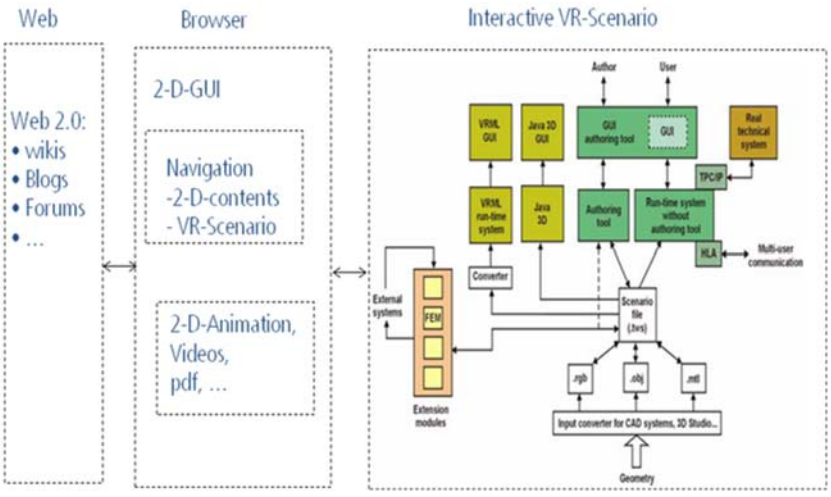


Fig. 4. Architecture of an integrated learning platform

free gas present in the insulating liquid, leaks insulating liquid and discharges the insulating liquid flow induced by a pressure wave in the direction of the conservator tank.

The training covering the Buchholz relay teaches trainees:

- What a Buchholz relays function in a transformator is
- How a Buchholz relay responds to a malfunction and
- How to replace/repair a Buchholz relay.

These three topics place different demands on the visualization and didactic treatment, requiring a flexible learning platform.

An integrated browser presents existing training materials such as operator manuals, 2-D animations and videos. A bidirectional connection between the VR scene and the browser contents allows systematically opening required information in the 3-D scene and additionally establishes a connection between the 2-D documentation (e.g. sectional drawing) and 3-D representation (see Figure 4). In addition, supplementary information such as user guidance can be presented formatted as a graphic and adapted to the client's corporate identity and easily used through common forms of interaction (e.g. links).

Practical training can be conducted with a flexible number of trainees, largely any time and any place. Both the schedule for exercises and the focus on individual work steps may be varied as desired. Errors do not have any negative consequences in the virtual scenario, what is equipment of the same design is being used, standardized operations can be applied anywhere without any "translation problems".

3 Summary

Virtual reality technologies have experienced a sizeable leap in development in recent years. Extraordinarily complex realities can be reproduced with the aid of VR. The Fraunhofer Institute for Factory Operation and Automation IFF is home to interactive high-level VR environments that can be specially applied in a broad range of industrial training programs ([cf.[2]). Both the basic technological and economic conditions will make broad use of interactive VR technologies in the basic and advanced vocational training of technical specialists possible in the near future. From the perspective of research, this is an impetus for research and development plans to intensify their focus on the potentials of learning in VR work environments. The technological developments presented here facilitate training on realistic virtual products, machinery and plants even when access to real objects, which are often not available for training at all or only to a limited extent, is limited. The use of VR systems in distributed learning environments is equally possible. The theoretical construct constitutes the foundation for researching the didactic and technical potentials of implementing VR systems and their potentials for education. A conceptual theory for research on learning actions in real and virtual technical systems is being worked on.

References

1. Arendarski, B., Termath, W., Mecking, P.: Maintenance of Complex Machines in Electric Power Systems Using Virtual Reality Techniques. In: IEEE 2008, Vancouver, Canada, pp. S483–S487 (2008) ISBN: 978-1-4244-2092
2. Belardinelli, C., Blümel, E., Müller, G., Schenk, M.: Making the virtual more real: research at the Fraunhofer IFF Virtual Development and Training Centre. Journal Cognitive Processing, S217–S224 (2008) ISSN 9217-224
3. Blümel, E., Jenewein, K.: Kompetenzentwicklung in realen und virtuellen Arbeitsumgebungen: Eckpunkte eines Forschungsprogramms. In: Schenk, M. (Hrsg.) Virtual Reality und Augmented Reality zum Planen, Testen und Betreiben technischer Systeme, pp. 177–182. IFF, Magdeburg (2005)
4. Martens, A., Bernauer, J., Illmann, T., Seitz, A.: Docs 'n Drugs - The Virtual Polyclinic - An Intelligent Tutoring System for Web-Based and Case-Oriented Training in Medicine. In: Proc. of the American Medical Informatics Conference, AMIA, Washington, USA, pp. 433–437 (2001)
5. Schenk, M.: Virtuelle Realität - Trends und Anwendungen für die Zukunft. In: Schenk, M (Ed.): Wettbewerbsvorteile im Anlagenbau realisieren. Zukunftsszenarien und Erfahrungsberichte. Tagung Anlagenbau der Zukunft. Magdeburg, pp. 97–103 (2006)
6. Schenk, M., Blümel, E., Schumann, M., Böhme, T.: Virtuelle Werkzeugmaschinen real gesteuert. wt Werkstatttechnik 96(7/8) (2006)
7. Schenk, M., Blümel, E.: Lernplattformen zum Anlauf und Betrieb von Produktionssystemen. Industriemanagement - Zeitschrift für industrielle Geschäftsprozesse, Heft 3, S23 – S26 (2007) ISSN 1434-1980

Evaluating Students' Concept Maps in the Concept Map Based Intelligent Knowledge Assessment System

Alla Anohina-Naumeca and Janis Grundspenkis

Department of Systems Theory and Design, Riga Technical University, Kalku Str. 1, LV-1658 Riga, Latvia
{alla.anohina-naumeca,janis.grundspenkis}@rtu.lv

Abstract. The paper presents results of the preliminary analysis related to the development of the evaluation mechanism for students' concept maps in the concept map based intelligent knowledge assessment system. Scoring schemes intended for human-based evaluation of concepts maps and embedded in computer-based concept mapping assessment systems are discussed. Overview of the developed system is given and scoring mechanisms implemented in its earlier prototypes are described. Factors affecting students' score in the mentioned system are identified and decisions concerning the development of a new evaluation mechanism are specified.

Keywords: knowledge assessment system, concept maps, scoring mechanism.

1 Introduction

Concept maps (CMs) as an assessment tool represent knowledge in form of a graph which nodes correspond to concepts in a domain, but arcs indicate relationships between concepts. Arcs can be directed or undirected and with or without linking phrases on them. A linking phrase specifies the kind of a relationship between concepts. The main constituent part of a CM is a proposition displaying a relationship between two concepts and corresponding to an elementary unit of knowledge. Usually CMs are represented in a hierarchical fashion [1] and a particular group of hierarchically related concepts is called a segment. Cross-links are relationships between concepts in different segments [2]. Various CM based tasks can be offered to students, however, two main groups of them are: a) "fill-in-the-map" tasks, where the structure of a CM is given to a student and he/she must fill it using the provided set of concepts and/or linking phrases, and b) "construct-a-map" tasks, where a student must decide on the structure of a CM and its content by him/herself.

The Department of Systems Theory and Design of Riga Technical University has been developing a CM based intelligent knowledge assessment system (KAS) since the year 2005. Four prototypes have been already implemented and experimentally evaluated [3]. The current development direction of the mentioned system is the elaboration of an automated scoring mechanism for evaluation of students' CMs. The paper describes evaluation schemes implemented in earlier prototypes of the KAS and presents results of the preliminary analysis concerning a new scoring mechanism.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 8–15, 2010.

The rest of the paper is structured as follows. Section 2 gives an overview of CM scoring schemes already proposed. Section 3 describes main functionality of the KAS. Section 4 focuses on evaluation of students' CMs in the earlier prototypes of the system, factors affecting students' score and decisions made regarding a new evaluation mechanism. Conclusions are presented at the end of the paper.

2 Related Works

Actually a great number of scoring schemes intended for human-based evaluation have been developed. Our analysis presented in detail in [4] shows that most of them are based on quantitative measures (number of valid propositions, levels of hierarchy, etc.) and only few combine both quantitative and qualitative (categorization of propositions according to the degree of their correctness) approaches. Considering quantitative measures it is difficult to evaluate if a student receives valuable information about his/her knowledge level when he/she is presented with such kind of data. Moreover, the greater part of structural scoring schemes are mainly applicable only for hierarchical CMs because such aspects as levels of hierarchy and cross-links are taken into account. Typically comparing students' CMs with one or more experts' maps a closeness index showing the extent to which the CM of a student matches that of the expert is calculated. The most schemes are developed for the evaluation of "construct-a-map" tasks which belong to the most difficult ones for the development of computer-based CM assessment systems. At the same time evaluation problems of much more simple "fill-in-the-map" tasks still remain open despite the fact that they can be easily embedded in computerized assessment systems and evaluated using an expert map and quantitative measures.

The known computer-based concept mapping assessment systems use rather primitive scoring schemes and in the best case validity of concepts and propositions in students' CM in relation to an expert CM is considered. One of the most advanced systems in this direction is COMPASS [5] offering a range of CM based tasks, performing quantitative and qualitative analysis of a student's map and identifying several categories of students' errors. Weights are defined by a teacher for each concept and proposition in a teacher's CM, as well as for each category of errors. So, the student's score is calculated as a similarity index taking into account concepts and propositions in an expert map and their correctness in a student's map.

Unfortunately, a number of important factors (for example, the level of task difficulty, the number of mistakes made at each level, the frequency with which students use provided help and feedback, etc.) are not considered at all in the examined systems. In our already implemented KAS the mentioned elements play an important role and must be taken into account.

3 Overview of the KAS

The KAS has twofold goals: a) to promote students' knowledge self-assessment, and b) to support a teacher in the improvement of learning courses through analysis

of results of systematic assessment of students' knowledge. The developed system is used in the following way [6]. A teacher defines stages of knowledge assessment and creates CMs for all of them by specifying relevant concepts and relationships among them in such a way that a CM of each stage is nothing else then an extension of the previous one. During knowledge assessment a student solves a CM based task corresponding to the assessment stage. After a student has submitted his/her solution, the system compares a student's CM with the teacher's one and generates feedback.

At the moment the system provides rich students' support (provided help and feedback) in comparison with other systems [7]. Three kinds of help are supported. Firstly, the system offers 3 "fill-in-the-map" tasks (Task1-insertion of concepts in the structure of a CM containing linking phrases, Task2-insertion of concepts in the structure of a CM without linking phrases, Task3-insertion of concepts and linking phrases in the structure of a CM) and 2 "construct-a-map" tasks (Task4-creation of a CM from the given set of concepts, Task5-creation of a CM from the given sets of concepts and linking phrases). Eight transitions between tasks are implemented allowing a student to find a task most suitable for his/her knowledge level. Four of them increase the degree of task difficulty and other four transitions reduce it. Secondly, in "fill-in-the-map" tasks a student can choose a concept from the given set of concepts and ask the system to insert it into the right place (node) within the structure of a CM. Thirdly, in all previously mentioned tasks a student can choose a concept from the given set of concepts and ask the system to explain it using one of the following types of explanations: definition, short description or example.

Feedback consists of numerical data (maximum score, actual student's score, total time for the task completion, time spent by a student), student's CM marked with labels representing his/her received points for each relationship and possibility to check a proposition. Checking of a proposition is supported at all previously described degrees of task difficulty. A student points out his/her created proposition and the system checks its correctness. Moreover, this feedback in case of incorrectness of a proposition presents explanations of both concepts involved in the proposition as it was described above.

4 Evaluation of Students' Concept Maps

A teacher's created CM serves as a standard against which students' CMs are compared in the KAS. Moreover, a comparison algorithm has been developed which is sensitive to the arrangement and coherence of concepts in students' CMs [8]. The algorithm is capable of recognizing different patterns of a student's solution. Two types of relationships are used in CMs: a) important relationships which show that relationships between the corresponding concepts are considered as important knowledge in a learning course, and b) less important relationships that specify desirable knowledge. For each correctly provided important relationship a student receives 5 points, but for each less important relationship only 2 points are assigned. Each proposition can be evaluated considering relative contribution of its parts: the presence of a relationship in a student's CM - 40%, a correct linking phrase - 30%, a

correct direction of the arc - 15%, a correct type - 10%, both concepts related by the relationship are placed in the correct places - 5%.

At the moment the system can recognize more than 36 different patterns of correct and partly correct propositions in students' CMs [8]. Recently the improvement of the mentioned algorithm was made by considering the so called "hidden" relationships in students' CMs which are nothing else than the derivation of relationships presented in a teacher's CM. The hidden relationships are correct too and could appear in students' CMs. They are scored by 1 point.

In general, students' CMs are scored by identifying how many correct relationships a student has defined:

$$P = \sum_{i=1}^{n} p_i * c_i \,. \tag{1}$$

where P is a student's score after the completion of a task, p_i is the maximum score according to the type of an i-th relationship, c_i is the coefficient that corresponds to the degree of the correctness of an i-th relationship, and n is the number of relationships in the CM structure including hidden relationships.

However, eq.(1) can be used only in case, if a student has completed a task without asking for help. However, taking into account that the system provides several kinds of students' support a correction mechanism must be applied in order to compare: a) results of those students who completed an original task without asking help and those who used help, and b) results of students who performed task at the higher difficulty degree and those who performed the same task at the lower difficulty degree.

Before considering development of the correction mechanism scoring schemes implemented in earlier prototypes of the KAS are described.

4.1 Scoring Schemes Implemented Early

In the year 2006 two versions of the system was implemented. Each of them supported different approach to the changing of the degree of task difficulty: insertion of additional concepts into a CM and offering of different types of tasks.

Thus, in the first approach a task of filling-in a teacher defined CM structure by a given set of concepts was offered to students. During the completion of a task a student could ask to reduce the degree of task difficulty. In this case the system inserted some concepts into the right nodes of the structure of a CM. Two main factors were taking into account developing a correction mechanism [10]: the number of difficulty reduction times and the number of concepts inserted by the system. It was necessary for two reasons. Firstly, concepts inserted by the system facilitated the further solving of a task. Secondly, before the reduction of the degree of task difficulty the system checked a student's solution and only correct or partly correct concepts remained in the CM, while other concepts were removed from it.

The correction coefficient was introduced. Its initial value was 1, but in the case of the reduction of difficulty this coefficient was decreased. The decrease consisted of two parts: a „penalty" for the number of the reduction times of the degree of task difficulty and the proportion of the number of additionally inserted concepts to the

total number of concepts within the map. Thus, a student's score was calculated combining eq.(1) with the correction coefficient [10]:

$$P = \left(\sum_{i=1}^{n} p_i * c_i\right) * \left(1 - c_s * s - \sum_{i=1}^{j} \frac{(a + \frac{\Delta * (i-1)}{m}}{m}\right). \quad (2)$$

where c_s is the penalty for each difficulty reduction time, s is the number of difficulty reduction times, a is the penalty for the insertion of the first concept by the system, j is the total number of concepts inserted by the system, m is the total number of concepts in a CM, and Δ is the increase of the penalty for each concept insertion.

In the second approach five tasks described in Section 3 and transitions between them were implemented. So, the previously specified eq.(1) was modified by a coefficient of the degree of difficulty for a given task in the following way [11]:

$$P = \sum_{i=1}^{n} lk_i * p_i * c_i \,. \quad (3)$$

where lk_i is the coefficient of the degree of difficulty for a given task. The coefficient lk_i was assigned to each relationship, but not to the whole task, because a CM of the current assessment stage could contain relationships defined at the previous stages on different degrees of difficulty. Assignment was made during the completion of a task and depended on the degree of task difficulty [11].

4.2 Factors Affecting Students' Score in the KAS

Improvement of the functionality of the system has lead to the necessity to develop an appropriate evaluation mechanism. However, several important issues appeared. In general, the score of a teacher's CM is calculated taking into account directly created relationships, where each important relationship is weighted by 5 points, but less important – by 2 points. Let's define this score as P_{norm}:

$$P_{norm} = 2 * x + 5 * y \,. \quad (4)$$

where P_{norm} is the score of a teacher's CM taking into account only directly created relationships, x is the number of less important relationships in a teacher's CM, and y is the number of important relationships in a teacher's CM.

However, as was pointed out before in Section 3, hidden relationships can be revealed in a teacher's CM and they may appear in students' CMs. As a result we can calculate the maximum score (P_{max}) for a teacher's CM:

$$P_{max} = P_{norm} + z \,. \quad (5)$$

where P_{max} is the maximum score of a teacher's CM taking into account directly created and hidden relationships, and z is the number of hidden relationships.

A student's score (P_s) will be equal with P_{max} in case if a student has related all concepts in the same way as they are related in a teacher's CM and has revealed all

hidden relationships. P_s will be equal with P_{norm} in case if a student's CM completely matches a teacher's CM, but he/she has not revealed hidden relationships. Here, the first question arises: How to compare results of students whose score is equal with P_{max} and those, whose score is equal with P_{norm}? Is the latter worse that the former? It is not the case, because he/she has the same structure of a CM as a teacher.

However, the both previously described cases are ideal cases and, as our experience shows, very few students can reach such results. Typically the greater part of student's relationships will be only partly correct taking into account direction of arcs, correctness of linking phrases, etc. Usually, students' score is less than P_{norm}. But in this case we have the next question: How to compare results of students whose score is less than P_{norm} but whose have hidden relationships in their CMs and students whose score is less than P_{norm}, but they do not have extra relationships?

Besides, not only the presence of hidden relationships affects the score of students. Other factors are related to the usage of help: a) students can reduce the degree of task difficulty during the completion of a task; b) students can ask explanations of concepts in all tasks; c) students can check propositions in all tasks, and d) students can ask to insert chosen concepts to right places in "fill-in-the-map" tasks.

Moreover, considering different degrees of task difficulty there is different number of penalties for partly correct propositions. Table 1 shows that working at the 4^{th} degree of task difficulty students always will receive the greater score because there are only 2 penalties, but at the 1^{st} and the 3^{rd} degrees always the less one. This is due to the fact that entry "no" means that for missing parts a student receives full points, for example, at the 5^{th} degree of task difficulty places of all concepts are considered as correct, but in reality places are not important for this task. A problem that arises from the different number of penalties is the following. Working at the 3^{rd} degree of task difficulty a student can place all concepts on correct places, but does not provide linking phrases, and after that to reduce the degree of task difficulty. After moving to the 2^{nd} degree he/she can submit his/her solution and receive the maximum score because linking phrases are not used at this degree.

4.3 Development of a New Scoring Mechanism

Considering the problems mentioned above the following decisions have been made. Firstly, two modes of system's operation must be provided: a) a mode of knowledge self-assessment which purpose is to allow a student to assess its own knowledge level and to learn more about a particular topic in case of incomplete or incorrect knowledge, and b) a mode of knowledge control intended for the determination of students' knowledge level by a teacher.

Secondly, a correction mechanism must be applied in different ways in each mode. During knowledge self-assessment the reduction of a student's score will not be performed in case of usage of such kinds of help as checking of a proposition or explanation of a concept. According to [7] both kinds of support provide not only help in task completion, but also tutoring. Thus, it is not correctly to reduce a student's

Table 1. Number of penalties at different degrees of task difficulty

Difficulty degree	An incorrect place of a concept	An incorrect type of a relationship	An incorrect linking phrase	An incorrect direction of an arc	Total number of penalties
5th	No	Yes	Yes	Yes	3
4th	No	Yes	No	Yes	2
3rd	Yes	Yes	Yes	Yes	4
2nd	Yes	Yes	No	Yes	3
1st	Yes	Yes	Yes	Yes	4

score after usage of such help in the mode of knowledge self-assessment. However, points must be reduced in case of additional insertion of concepts because this kind of help substantially facilitates the task completion particularly if fundamental concepts are inserted. Moreover, it is necessary to avoid situations when a student in such a way inserts all concepts and receives the maximum score. In the mode of knowledge control all kinds of help will contribute to the reduction of a student's score.

Thirdly, it is necessary to define the restriction on the maximum number of propositions which can be checked (only for the mode of knowledge control) by a student and the maximum number of concepts which can be inserted by the system if a student asks for it. Two approaches must be implemented: a) the number of checking allowed must be calculated automatically by the system taking into account the total number of propositions or concepts for a task of the current assessment stage, and b) a teacher's possibility to change the restriction automatically set by the system.

In case of explanation of a concept a student's score must be reduced only in case if a student asks explanation of a certain concept for the first time. After that he/she can receive explanation of the same concept repeatedly without reducing the score.

In the mode of knowledge control penalty for the usage of help must grow each time when a student uses help in order to stimulate his/her to complete a task by him/herself and to use the reduction of difficulty as seldom as possible.

Fourthly, considering a problem related to hidden relationships we decided to provide additional feedback both to a student and to a teacher by showing P_{norm}, P_{max}, P_s, number of hidden relationships in a teacher's CM and revealed by a student.

5 Conclusions and Future Work

An appropriate evaluation mechanism of students' CMs is an important part of any computer-based concept mapping assessment system. Regardless of the fact that a lot of CM scoring schemes have been developed the greater part of them are intended for human-based evaluation of "construct-a-map" tasks. As a result, their feasibility and usefulness in CM based knowledge assessment systems is a discussible question. In turn, evaluation mechanisms implemented in the known assessment systems do not consider such important factors as the level of task difficulty, the number of mistakes made at each level, the frequency of used help, etc. In the authors' developed KAS the following factors affect students' score: the presence of hidden relationships in students' CMs, possibility to reduce the degree of task difficulty, three kinds of help

available and different number of penalties at each degree of task difficulty. As a result, the necessity to develop an appropriate evaluation mechanism has arisen. So far, the following decisions have been made: it is necessary a) to provide two modes of system's operation (knowledge self-assessment and knowledge control) and to apply the correction mechanism of students' score in different ways in each mode, b) to implement two approaches (automatic calculation by the system and changing by the teacher) regarding the restriction on the maximum number of propositions which can be checked by a student and the maximum number of concepts which can be inserted by the system, c) to reduce the students' score only in case if a student asks an explanation of a certain concept for the first time, d) to provide growing of penalty for the usage of help in the mode of knowledge control in order to stimulate a student to complete the task by him/herself, and e) to provide additional feedback to a student and to a teacher concerning hidden and directly created relationships. Future works is related to the development of a mathematical model for scoring CMs in the KAS.

References

1. Novak, J.D., Cañas, A.J.: The Theory Underlying Concept Maps and How to Construct and Use Them. Technical report, Florida Institute for Human and Machine Cognition (2008)
2. Cañas, A.J.: A Summary of Literature Pertaining to the Use of Concept Mapping Techniques and Technologies for Education and Performance Support. Technical report, Pensacola (2003)
3. Grundspenkis, J., Anohina, A.: Evolution of the Concept Map Based Adaptive Knowledge Assessment System: Implementation and Evaluation Results. In: 49th Scientific Conference at Riga Technical University, pp. 13–24. RTU Publishing, Riga (2009)
4. Anohina, A., Grundspenkis, J.: Scoring Concept Maps: an Overview. In: 10th International Conference on Computer Systems and Technologies, Ruse, Bulgaria, pp. IV.8-1–VI.8-6 (2009)
5. Gouli, E., Gogoulou, A., Papanikolaou, K., Grigoriadou, M.: Evaluating Student's Knowledge Level on Concept Mapping Tasks. In: 5th IEEE International Conference on Advanced Learning Technologies, Kaohsiung, Taiwan, pp. 424–428 (2005)
6. Vilkelis, M., Anohina, A., Lukashenko, R.: Architecture and Working Principles of the Concept Map Based Knowledge Assessment System. In: 3rd International Conference on Virtual Learning, Constanta, Romania, pp. 81–90 (2008)
7. Anohina, A., Grundspenkis, J.: Student's Support in the Concept Map Based Knowledge Assessment System. In: 7th European Conference on e-Learning, Agia Napa, Cyprus, pp. 38–45 (2008)
8. Anohina, A., Vilkelis, M., Lukasenko, R.: Incremental Improvement of the Evaluation Algorithm in the Concept Map Based Knowledge Assessment System. Int. J. Comp., Com.& Cont. 4(1), 6–16 (2009)
9. Grundspenkis, J., Strautmane, M.: Usage of Graph Patterns for Knowledge Assessment Based on Concept Maps. In: 49th Scientific Conference at Riga Technical University, pp. 60–71. RTU Publishing, Riga (2009)
10. Anohina, A., Lavendelis, E., Grundspenkis, J.: Concept Map Based Knowledge Assessment System with Reduction of Task Difficulty. In: 16th International Conference on Information Systems Development, Galway, Ireland, pp. 853–866 (2007)
11. Anohina, A., Pozdnakovs, D., Grundspenkis, J.: Changing the Degree of Task Difficulty in Concept Map Based Assessment System. In: IADIS International Conference e-Learning 2007, Lisbon, Portugal, pp. 443–450 (2007)

Agent-Based Simulation Use in Multi-step Training Systems Based on Applicant's Character Recognition

Ieva Lauberte, Egils Ginters, and Arnis Cirulis

Sociotechnical Systems Engineering Institute, Vidzeme University of Applied Sciences,
Cesu Street 4, LV-4200 Valmiera, Latvia
{Ieva.Lauberte,Egils.Ginters,Arnis.Cirulis}@va.lv

Abstract. One of the tasks of intelligent education technologies is intensification of a training process that can be achieved using different and advanced Information and Communication Technologies and Electronic (ICTE) tools. Nevertheless, important changes in training process are necessary, because simple use of ICTE benefits will not promise a good success. Trainees have different perception of information, which determines the demand for various forms of visual presentation of learning material and diverse style of the training session. To recognize the type of the character and mode of perception of a trainee the authors of the article offer agent-based simulation model TemPerMod operating in NetLogo environment.

Keywords: Agent-based simulation, NetLogo, Temperament, Perception, e-learning.

1 Introduction

Nowadays, the Sociotechnical Systems Engineering institute works on the new e-learning technology based on introduction of virtual and augmented reality (VR/AR) solutions and simulation. The technology foresees the training of a trainee in conformity with individuality of his perception [1]. The training model (see Fig. 1) provides splitting the training cycle in the set of operations in conformity with the scenarios elaborated earlier. However, each of the scenarios and operations respects the type of personality and individuality of perception of a trainee.

Although the training process begins with the base scenario, always after the fixed time slot the feedback from the simulation model of the trainee is received. It is used for checking the quality of the obtained skills necessary for the implementation of each technological operation.

The model recommends (with the calculated probability) to continue the current scenario or switch to other more suitable scenario. Further the training material continues from the same or other technological operation.

The training material is implemented using VR/AR tools allowing reducing training costs by replacing expensive technological equipment. Sometimes it is also claimed by the rules of labour safety.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 16–22, 2010.

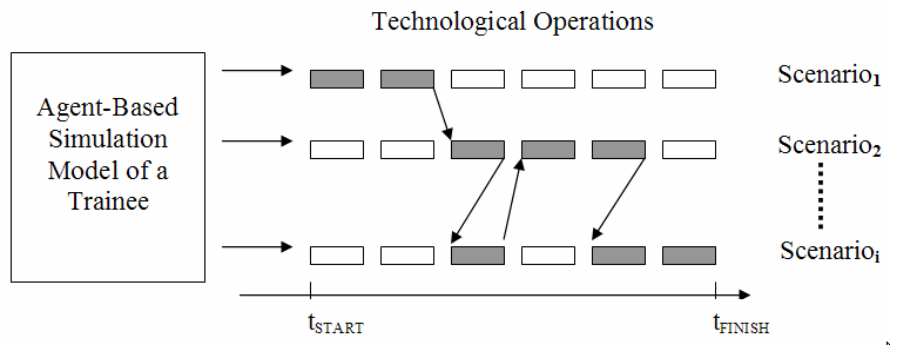


Fig. 1. Multi-step training process based on scenarios generation

2 Training Scenarios Generation

Each training $Scenario_{TR}$ belongs to the set of generated scenarios $Scenario_{TR} \in \langle Scenario_i \rangle$, where $i = 1, N$, and N – the total amount of generated scenarios. Each scenario consists of the set of training operations (steps) (see Fig. 1) $Scenario_i = \{Op_i^j\}$, where $j = 1, K$, and K – the total amount of operations involved in $Scenario_i$.

Any e-learning system is sociotechnical. It is appropriate combination of logical and physical structures (see Fig. 2) where logical structure determines the essence and identity of the e-learning system, but physical structure is tangible part of the goal system.

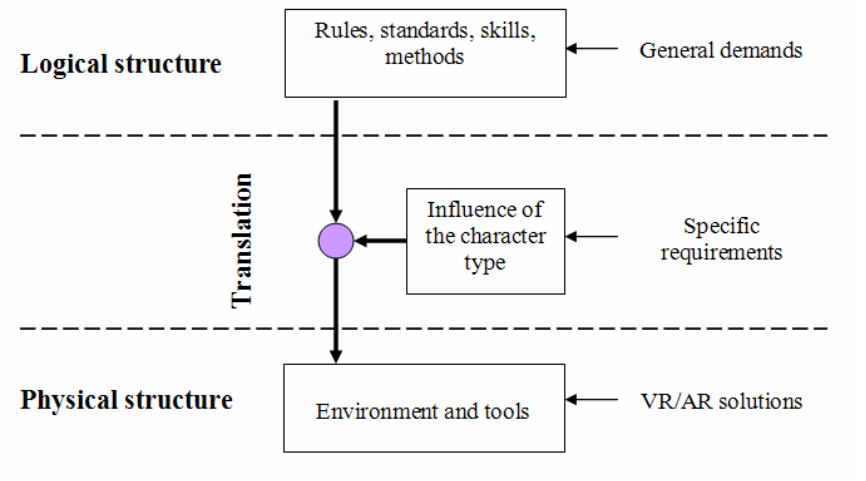


Fig. 2. Training scenarios generation

The logical structure involves rules, standards, skills, methods etc. necessary to implement predefined work operation. Otherwise, physical structure is always determined by logical structure and comprises technological equipment, software and hardware. To reflect (translate) the logical structure to physical the specific demands determined by the character of trainee must be respected, which causes some perturbation in translation.

Visualisation form and features of each training operation $\{Op_i^j\}$ depend on the type of character of trainee bringing specific requirements to the training process.

The result of the training will depend on how precise will be determined the type of the character and perception mode of the trainee.

3 Agent-Based Character Recognition

Every person is unique at least with own reactions to various situations, because people have different types of personality and temperament.

The most popular classification of temperament was introduced by Greek physicians Galen's and Hippocrates [2]. The four basic temperament groups $\langle T \rangle$ are sanguine, phlegmatic, choleric and melancholic. Those temperaments were later discussed by Kant [2], but Pavlov correlated the types of higher nervous system with psychological types of temperament [3, 4]. Sanguine persons $\{t_1\}$ are very active and communicative. They know how to make good relationships [5]. Phlegmatic person $\{t_2\}$ is calm, balanced and emotions have no essential importance in his life [5]. They do not put so high importance to relations with other people and their appreciations. Choleric person $\{t_3\}$ has a lot of ambition, energy, and passion [6]. They are more impulsive as others and have all characteristics to act as a leader [5]. Melancholic persons $\{t_4\}$ are more sensitive than other. Melancholic persons have rootless feeling from inside, instability, which expresses like indecision [5].

Other classification was done by Keirsey [7, 8], who introduces the new types of temperament: artisan, rational, idealist and guardian, more emphasising professional suitability of the person. Later four temperament colours $\langle C \rangle$ according to temperaments [9] were proposed:

- Sanguine – Yellow $\{c_1\}$ (the colour of adventure and artistry);
- Phlegmatic – Blue $\{c_2\}$ (the colour of cold clear: logic and perception);
- Choleric – Magenta $\{c_3\}$ (the colour of intuition and transcendence);
- Melancholic – Red $\{c_4\}$ (the colour of authority and stability).

Later Grey colour $\{c_5\}$ was added by Dellinger [10].

In late 90-ties Dr. Susan Dellinger [10] introduced the term psychogeometrics and explained not only how to determine your own personality type, but how to use geometric psychology to identify the beliefs, values, and attitudes of any person you meet. She considered that each shape having specific form (circle, triangle, square, squiggle, and rectangle) represents a personality, and believed that five personalities are within us, but that we have one dominant personality and one secondary personality that we use the most.

According to Dellinger, 83% of the time the shapes $\langle F \rangle$ you have just chosen will accurately represent your primary and secondary personalities [10]. Circle $\{f_1\}$ represents harmony, unity and balance. Persons are good team players and communicators, best listeners, empathetic and sensitive to others' needs. Box $\{f_2\}$ represents the structure. A person is tidy, logical, practical, focused and detailed. A person is resistant to change and does not "natural" team player. Triangle $\{f_3\}$ represents ambition and high achievement. They are goal setters, high achievers and decision makers. Rectangle $\{f_4\}$ represents changes and inner growth or transition. A person is temporary and not certain about the future. Squiggle $\{f_5\}$ represents high energy, animation, sense of humour and creative intelligence. Powerful like the Triangle, and does not good team players.

The activity $\langle Act \rangle$ of the person is attributing of each type of the personality. Activities can be measured from static to very active and even chaotic movement.

The most important factor in training process is perception $\langle P \rangle$ of the trainee. In conformity with the ideas of Bandler and Grinder (70-ties) related with Neuro-Linguistic Programming (NLP) [11] at least three types of perception exist. Visual $\{p_1\}$ person perceives information in image form. Auditory $\{p_2\}$ person perceives information by ear, but Kinaesthetic $\{p_3\}$ is a person with predominant sense. Of course, it is most difficult case for training.

To determine which type of temperament the person has and which style of learning is more suitable, many tests would be taken. For instance, Jung Typology test [12] consists of 72 questions, where each question has only two possible answers - "yes" or "no". Test result is like letter formula according to Carl Jung and Isabel Myers-Briggs typology along with the strengths of the preferences.

The Keirsey Temperament Sorter (KTS-II) [7] is based on Keirsey temperament theory mentioned above. Test consists of 71 questions with two possible answers. The Gray-Wheelwright Winer 4-letter Type Indicator test [13] consists of 70 questions with two possible answers. Testing results are compatible with Myer Briggs and Keirsey tests. Soloman and Felder "Index of Learning Styles Questionnaire" [14] test consist of 44 questions with two possible answers. This test is compatible with four dimension Felder-Silverman Learning style model. Chislet and Chapman "VAK Learning styles Self-Assessment Questionnaire" [15] test consists of 30 questions with three possible answers. This test can determines learning style of trainee in conformity with NLP definitions.

The tests mentioned above are widespread, but they are complex, asking for well prepared trainee, and they are based on trainee opinions about oneself, therefore give different results depending on person's mood.

The authors offer to use agent-based approach. An agent-based simulation technology is not new, and it is used in social sciences for a quite long time. The term „agent-based simulation" refers to a particular type of simulation, which has two essential components – agents and environment [16]. An agent's behaviour is the result of rules predefining interactions among them and environment. The environment has certain autonomy, but it can also be influenced by the agents' behaviour. Agent-based simulation models can be used for specification the complex, dynamic, and interactive processes that exist in the real world [17]. The benefit of agent-based simulation is

possibilities of estimation the all important factors simultaneously reducing the chance of the trainee to manipulate with the answers.

In agent-based model for character recognition TemPerMod the following background is used:

- Galen's and Hippocrates (personality types) [2];
- Kersey's taxonomy (temperament types) [7,8];
- Susan Dellinger (psychogeometric) [10];
- Bandler and Grinder (NLP) [11].

As mentioned above, trainees can be classified by the type of personality or temperament $\langle T \rangle$. Each group has appropriate behaviour or activity $\langle Act \rangle$, corresponds to predefined colour $\langle C \rangle$ and the favourite form of the objects $\langle F \rangle$. We can suppose that activity, colour and form combine the set of attributes $\langle A \rangle$. One more factor very important for successful training is perception $\langle P \rangle$. Therefore the goal of the agent-based simulation model TemPerMod is recognizing the type of the personality $\langle T \rangle$ and the kind of perception $\langle P \rangle$:

$$\langle T, P \rangle \leftarrow A(C, F, Act) \tag{1}$$

Complete testing process consists of two parts:

- Audio-visual modelling game based on agent-based model TemPerMod;
- Interview with the trainee about his impressions about the game.

The TemPerMod is written in NetLogo [16]. The modelling desktop has the form of a pentagon. Pentagon is divided in five frames, where each frame initially involves equal amount of the agents having the same form $\langle F \rangle$ and the same colour $\langle C \rangle$. Inside of the desktop is smaller pentagon. Each frame has the door to the inner pentagon. Agents in their frames move with different speed of motion $\langle Act \rangle$. During the simulation cycle agents can get in the inner pentagon. Step by step the inner pentagon fills with different agents. When inner pentagon is completed then first cycle is finished. The agent colour, form or activities are not critical for filling the inner pentagon that is random process. During the simulation cycle also some audio information is announced. Some information can be depicted on the desktop also in written form. Modelling cycle continues 8-10 seconds. After 2 seconds the next cycle begins. At the end of simulation succeeds the interview with the trainee about his impressions from the simulation game int. al. what kind of colour, form or activity of the agents are most likeable.

The TemPerMod model was validated by Jung Typology test [12] as one of most popular. The validation results revealed that the results of the character recognition match for 85%. Who is more precise Jung or the authors of the article this is the task for further research.

4 Conclusions

Nowadays, the transition from the traditional learning in classrooms to on-line training and consulting is going on. However, obstacle that trainees have different perception of information and type of character often does not respected. Those differences determine specific requirements to visualisation of learning objects and the style of presentation. To recognize the type of the temperament of a trainee the testing is implemented. The authors of the article offer agent-based simulation model TemPerMod for recognition of the temperament and perception of a trainee.

The TemPerMod model is created in the NetLogo environment, and like to other multi-agent based simulation systems asking for serious computing resources. Therefore, the model further must be improved focusing on reducing required modelling session time and resources, and adding some more parameters for achieving higher accuracy in character recognition.

Introduction of TemPerMod in e-learning systems will promote reducing the training time and necessary funding assigned.

References

1. Ginters, E., Cirulis, A., Akishin, V.: Virtual environment use in e-learning. In: Proceedings of 6th WSEAS International Conference on E-Activities 2007, Puerto de la Cruz, Tenerife, Spain, December 14-16, pp. 12–17 (2007) ISBN 978-960-6766-22-8
2. Chamorro-Premuzic, T.: Personality and Individual Diferences. BPS Blackwell, Malden (2007)
3. Barteneva, D., Reis, P.L., Lau, N.: Bilayer agent-based model of social behaviour: how temperament influence on team performance. In: Proceedings 21st European Conference on modelling and simulation (2007)
4. Barteneva, D., Reis, P.L., Lau, N.: Implementation of emotional behaviours in multi-agent systems using fuzzy logic and temperamental decision mechanism. In: Proc. EUMAS 2006: Fourth European Workshop on Multi-Agent Systems, Lisboa, Portugal (2006)
5. Renge, V.: Psihologija. Personibas psihologija (in Latvian) Riga, Zvaigzne ABC, 16–29 (2000)
6. Wikipedia: Four temperaments (2008), http://en.wikipedia.org/wiki/Four_Temperaments
7. Keirsey.com: The Keirsey Temperament Sorter, KTS-II (2008), http://www.keirsey.com/sorter/register.aspx
8. Keirsey, D.: Please Understand Me II: Temperament, Character, Intelligence, 1st edn. Prometheus Nemesis Book Co. (1998) ISBN 1885705026
9. Religa, J.R.: Temperament Colors System (2008), http://www.jedigirl.com/www/personality_types/temperament/index.html
10. Dellinger, S.: Communicating Beyond Our Differences: Introducing the Psycho-Geometrics System, 2nd edn., Jade Ink (1996) ISBN 978-1892762009
11. Bandler, R., Grinder, J.: Frogs into Princes: Neuro Linguistic Programming, Moab. Real People Press, UT (1979)
12. HumanMetrics: Jung Typology Test (2008), http://www.humanmetrics.com/cgi-win/JTypes2.asp

13. Robert Winer, M.D.: The Gray-Wheelwright-Winer 4-letter Type Indicator Test (2009), http://www.neurocareusa.com/GWtest/GrayWheelwrightWiner4letterTestwAnswerSheet.pdf2006
14. Soloman, B.A., Felder, R.M.: Index of Learning Styles Questionnaire (2008), http://www.engr.ncsu.edu/learningstyles/ilsweb.html
15. Chislet, V., Chapman, A.: VAK Learning Styles Self-Assessment Questionnaire (2009), http://www.businessballs.com/freematerialsinword/vaklearningstylesquestionnaireselftest.doc2005
16. Lauberte, I.: Using cellular automata in agent-based simulation for regional development. In: Bluemel, E., Ginters, E.V. (eds.) Annual Proceedings of Vidzeme University College: ICTE in Regional Development, pp. 99–104 (2005) ISBN 9984-633-01-2
17. Smith, R.E., Conrey, R.F.: Agent-based modeling: a new approach for theory building in social psychology. Personality and Social Psychology Review 11(1), 87–104 (2007)

Application of Information Technologies to Active Teaching in Logistic Information Systems

Andrejs Romanovs, Oksana Soshko, Arnis Lektauers, and Yuri Merkuryev

Department of Modelling and Simulation, Riga Technical university,
1 Kalku Str., 1658, Riga, Latvia
{rew,oksana,arnis,merkur}@itl.rtu.lv

Abstract. Information Technology has always been a popular choice among high-school graduates when deciding on a field of study. Despite the comparatively high education levels among Latvian employees, there is still a lack of knowledge and practical skills crucial for competitiveness in a market based economy. In order to ensure relevance of the qualifications and adaptability in the fast changing environment, active learning and teaching have a special importance. Recent developments in information technology call for a serious reconsideration of the actual teaching methods and provide opportunities for developing a new educational methodology. The current paper focuses on application of IT within the course of logistics information systems for developing student practical skills and abilities. The necessity for an active teaching and learning e-environment is highlighted, and a concept of its realisation based on Web 2.0 technologies is discussed within LIS.

Keywords: active teaching and learning, Information Technology, logistics information systems, web 2.0.

1 Introduction to the Curriculum of Logistics Information Systems

The symptoms of necessity for the course Logistics Information Systems (LIS) in the Master Curriculum on Information Technology were pointed out firstly during participation in the European project „INCO Copernicus AMCAI 0312 (1994 – 1997) *"Application of Modern Concepts in the Automated Information Management in Harbours by Using Advanced IT – Solutions"*". The project's results showed a great lack of logistics specialists having efficient knowledge in information technology [1].

The course of LIS was developed for the post graduate students of the Department of Modelling and Simulation in 1998 by Professor Egils Ginters and Professor Yuri Merkuryev. The course curriculum became an outcome of a project LOGIS LV-PP-138.003 "Long-distance tutorial network in "Logistics Information Systems" based on WEB technologies" (2000-2002) [2]. The LIS course is aimed at providing students with high level knowledge, skills and competencies in Logistics Information Systems through the integration of theory and practice. The course focuses on the application of information technologies to logistics management.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 23–30, 2010.

The LIS course consists basically of postgraduate students having an average age of 22-24. Almost 90% of LIS students are employed either in private companies or in government organisations, which makes them to be in extremely high demanded for qualitative learning and teaching processes. Most of the students work in the IT field, which gives them deeper professional skills. For that reason, lecturers need to be able to adjust course material to suit students experience and prior knowledge.

The course is structured in several blocks. It starts with a course overview. According to the first principle of andragogy which states that, as the learner's need to know why learning is important and how learning will be conducted, the course structure, goals, outcomes and requirements must be discussed first. Moreover, a response to IT professional standards should be provided underlining the role of the LIS curriculum for getting a professional diploma. This is normally done in interactive discussion sessions, if the number of students is not too great. Finally, the lecturer outlines the course structure, its goals and outcomes.

The second block of the LIS course covers the main topics. The sequence of them through the course is not precisely defined, and is flexible to any lecturer/students requirements. Along with this theoretical block, students should improve their practical skills performing several tasks during labs, namely “GPS and GIS application for object positioning monitoring”, “Cargo Tracking Systems Analysis”, and “Radio Frequency Technology applications in Logistics”.

The next block of the LIS is aimed at both exploring and introducing the variety of information systems in the context of logistics management. Several solutions are discussed in the fields of transportation logistics, inventory management, warehouse logistics, production etc. In each case the focus is on the functionality of the system for supporting related logistics functions. However, despite exploring the functionality, technical solutions are also discussed in order to underline the correlation between information technologies and information system. In parallel with lecturer’s (and invited industrial partners as well) presentations, students make their own presentations of different logistics information systems. This task is performed as team-work and is aimed at both enhancing students’ professional competence and their group working skills. The block is finalised by evaluation tasks.

The student evaluation process is a critical challenge for every academic course. It should be realised in a way which: (1) allows adequate evaluation of student knowledge; (2) is effective for learning and in fact is a part of learning; and (3)covers students personal character traits (for example, some of them perform better on tests, some benefit more in oral examinations, others do better writing essays).

Initially, the evaluation of students was conducted at the end of the course and was organised as an examination. However, the main shortcoming of this is that examination at the end of the course usually leads students to postpone their studies to a few days before the exam. To improve the evaluation by making it an assessment-for-learning, in 2009 a new evaluation system was implemented. This can be called a portfolio assessment, in which students gather artefacts that illustrate their development over time. The evaluation portfolio in the LIS course consists of:

- An on-line test with 60 questions which covers the block of Logistics IT;
- Written essays on three questions in the context of block LIS;
- Team-work and lecturer presentations of the LIS.

Although, there are still some shortcomings in the current evaluation, the new way of assessing students has following benefits:

1. to motivate students to study during the course;
2. to minimise psychological stress during the assessment, by providing a possibility to improve the grade during next evaluations;
3. to provide a variety of assessment methods way for students. This is an essential point for discussion in a pedagogical context, because there is not just one 'best' way of examining the students. Some of them being "slow-thinkers" would feel a lot of pressure due to time limitations during the test. Others might feel more comfortable going deeply into the subject, and some like to give direct answers to precisely-defined questions;
4. to support both individual student work (and responsibility for the outcome) and team-work (where the responsibility for the evaluation is spread among all team workers).

The evaluation portfolio components may have differential weights which can be easily up-dated by the lecturer before the course is started.

2 Pedagogical Notes in LIS

There is little difference between the terms of 'teaching' and 'learning' in the current paper, however some differences still exist. 'Teaching' is explained as a part of educational process, where an active position (or role) is taken by a lecturer who presents (teaches) some material to students using different methods. Illustratively, a didactic lecture is a trivial method of teaching. In contrast 'learning' can be explained as a part of an educational process, where students actively construct their own knowledge by absorbing, understanding and analysing information provided by the lecturer. We can assume, that the student's role in teaching is more passive compared with the lecturer's, however learning is driven more by students (with some support and coordination from the lecturers side).

In LIS, the main focus now is on supporting students in active learning and, if possible, in student-centered learning. By active learning we understand "instructional activities involving students doing things and thinking about what they are doing". Active learning is the idea that different people learn in different ways. Understanding how learning can be realised, which is the better method of learning for each student and to provide different learning styles for students is one of the pedagogical objectives of LIS. Teaching aids are presented by text books, slide-show presentations and different video materials, etc.

Every teaching process consists of three components: students, teacher, and an environment. In teaching, the role of the lecturer is dominant and usually performed by a trainer (instructor, lecturer). In learning, the main components are students and the learning environment. The lecturer's function here is to support students with a variety of methods, tools, and environments. In this section, we consider the LIS course audience and discuss some teaching methods and tools.

Despite plenty of traditional didactical teaching aids, the actual focus now is on improving the quality of educational process applying different IT solutions. RTU academic personnel point out the great importance of using modern technologies in teaching. Illustratively [3, 4, 5, 6] describe the application of IT solutions in developing effective e-learning and evaluation methods.

In our experience, a lecturer must organise the course providing a balanced learning experience using different learning methods, *i.e.* lectures, labs, discussions etc., see Fig. 1. To illustrate, during typical classroom lectures, conceptual and theoretical information (intuitive learning) should be supplemented with concrete, practical information (sensory style) expressed through lecturers comments and explanations. Pictures, and diagrams of slides presented to visual learners must also be explained orally for verbal learners who seek explanations having words. Active learners prefer to do physical experiments and to learn by expressing themselves working in groups. They appreciate conducting lab exercises which can promote the students cognitive activities. For reflective learners, however, we provide tasks, such as evaluating different options, making analysis (of data acquired in Lab 1).

LIS, the most used teaching method, uses traditional lectures. This can be called a passive teaching method, where the lecturer has the main role. Lectures are used mostly in the Logistics and Information Technologies block, however it still has some active learning elements such as debriefing, discussions, and 5-minute activities done in pairs. The Logistics Information System block is organised using workshops, seminars and team-projects. Here, both lecturers and students have active roles, so this block can be characterised as an active learning support block.

Laboratory exercises are traditional method of active learning. Labs can be used to facilitate the exploration and illumination of difficult concepts. Most importantly, labs can enhance the cognitive learning process, which is often referred to as the integration of theory with practice.

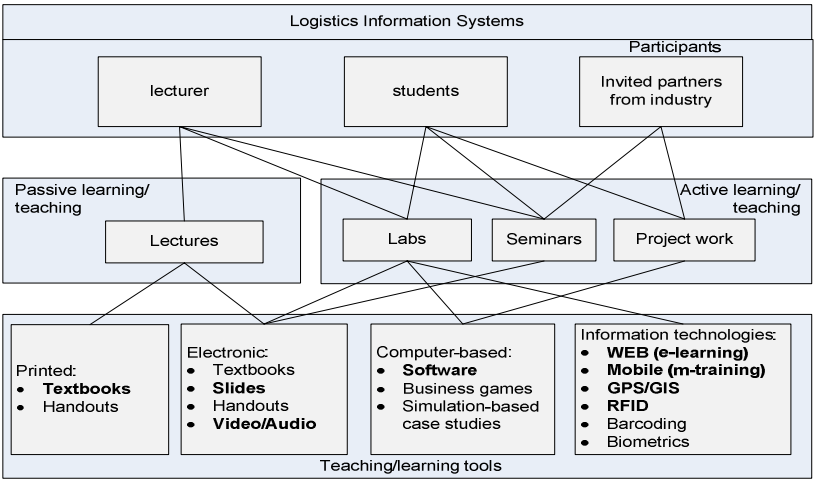


Fig. 1. Teaching components in LIS

In fact, information technologies within LIS are not only the subject of teaching, but rather a part of didactical tool aimed at demonstrating the power of IT in every field of application, such as logistics, education, entertainment and others. The possibility to learn information technologies/systems by applying them in studies allows students (1) to understand the main principles of IT in Logistics (which is the aim of the course), and (2) to evaluate the variety of its applications for different solutions (which is the outcome of the course). This, according to Bloom's Taxonomy of Educational Objectives, can be explained as student growth through development of their intellectual skills and abilities.

3 Teaching / Learning Environment in LIS

Application of modern IT for teaching/learning purposes in LIS started firstly with using on-line test as evaluation. Application of IT as an environment started in 2005, when all course materials were located at Moodle system. In 2008, ORTUS was created as a single electronic educational environment of RTU. Despite plenty of benefits being provided by these solutions, there are still some shortcomings as:

- it serves more as a storage for keeping such teaching material, as slides, video/audio, handouts etc. The active processes are forums and on-line tests.
- it doesn't allow the use of objects, to be placed inside the educational modules repeatedly; or the use of materials, created by lecturers and students, to remain in a programme, so that in the course of time a student loses access to them.
- it is not possible to organise a comfortable educational space for active co-operation between students working on team solutions to educational tasks.

All the above mentioned lead to the final conclusion concerning the necessity of re-designing the e-learning environment in order to satisfy the following requirements for active learning:

1. to implement active learning tools and methods;
2. to maintain learning activities like communication, discussions, team-working;
3. to support both collaborative and co-operative learning;
4. to support the lecturer's role as an active participant and co-ordinator, rather than as promoter;
5. to allow students to create valuable cooperative, collaborative and individual products, which later can be used by students in their professional carriers.

4 Application Tendencies of Modern IT in Teaching

When analysing applications of IT for teaching purposes, it is possible to select a number of influencing facts. In our opinion, the most important of them are the increased amount of information related to the permanent development of technologies and entrepreneurial activity, and the mass introduction in teaching the technologies on the basis of the Internet, including Web 2.0. The first factor causes changes in knowledge of the specialisation, and determines the main requirements of the environment for teaching, the rapid transmission of knowledge and abilities of the students. The

second factor allows an increase of the creative potential of students, provides joint creation and use of information resources and collaboration and expansion of functional possibilities for teaching. Contemporary students want teaching in the form of an active dialogue and to be in a position to have an impact on the course of events, that is, they want to be competent participants in a teaching process, the authors and reviewers, as a student-centered model foresees it.

Following the O'Reilly [7] definition, Web 2.0 is the technology for designing systems which, through network co-operation continue to develop as more people use them. The main feature of Web 2.0 is to attract the users to create knowledge by introducing home page content and to use the principle of frequent verification. In the base variant of Web 2.0, every person could easily create and spread *content* in the World Wide Web. It could include records in weblog, pass video through YouTube, place pictures in Flickr, help in the creation of content in wiki, and also create Myspace type social networks. Thus, contrary to Web 1.0, which makes service to a vertical «Teacher-Student» relationship, the Web 2.0 technology is characterised by development of horizontal connections and works on the basis of social relations.

The key components of Web 2.0 are easy-to-use instruments and general or social relationship systems having the expected results. One of the most interesting results of the use of Web 2.0 is the phenomenon which is often named *'Collective Intelligence'*, describing the situation when the potential influence of information between the users of WWW grows very quickly. It is very important, that this index increases with the increase in the number of persons actively contacting each other through WWW, that provides people with possibility of joint search, creation and the exchange of information. The research of McKinsey [8], Forrester [9] and other authors in the area of the use of Web 2.0 technology in industry, shows recently growing active interest in *Collective Intelligence*. Researchers point out that for effective creation of new decisions and knowledge it is necessary to aggregate the possibilities for users of the input of information, methods of joint activity, and also modern technologies of collection and processing of information (wiki, weblogs, widgets, mashups etc.)

5 Concept of e-Environment in LIS

As a result of research, the main conceptual requirements were formalised for the developed environment of e-learning with the use of modern approaches and information technologies. It is necessary to provide the e-environment having the following basic functionality:

- permanent development of educational materials, with the possibility of their modernisation by authors and teachers and by students. Traditional electronic courses serve only as base information sources;
- generalisation of existing knowledge and the creation of new knowledge – students create materials themselves and communicate with other students through technologies, enabling the distributed creation of materials and division of responsibility in the process of forming and the use of resources;
- use in the process of teaching large sections of the aggregated information sources which includes in itself all possible formats of files and methods of their transmission;

- the study of materials takes place at any time and at any place: all information sources can be used not only by computers but also by mobile telephones, MP3 players etc.

Development of these new possibilities for e-learning environment will be based upon the instruments of Web 2.0 technology, such, as weblogs, wiki, podcasts and other.

Articles written on weblog technology form an analogy of the classical concept of scientific theses and create electronic home pages of a persons or organisations on which the collection of information is made on a concrete topic or topics, including regular updates of this information. Information can be written down in a weblog by a proprietor. It can be re-written from other weblogs. The readers of a weblog can also supply information**,** make comments on themes and discuss different questions. Automatic creation of templates is thus possible for theses published in a weblog, using information from the pages of wiki and personal notes associated with them. A weblog can be integrated with other weblogs. The results of continuing experiments, current results of work and newly synthesised ideas, can be written down in a weblog.

It is possible to select the different forms of weblogs for the teaching of LIS, firstly as a means of communication between students concerning organisation of the course, the performance of tests and home tasks, and the support of different student initiatives, secondly, for additional discussion of course themes, conducted by a teacher and the encouragement of students to make independent analyses of the information received. In such weblogs, teachers will formulate questions and tasks for students, and also give references on additional materials and resources for the topic. Thirdly, for the students, using weblog on a research theme can become the method for bringing in comates and teachers to make comments, and to criticise and correct the method of preparation.

The addition of the use of weblogs for teaching LIS can be a forum – a traditional asynchronous mechanism of communication. A forum can be related to any theme, document, person, or weblog. A forum provides bilateral connections and enables comments both on the theme and the comments of other users.

For the personal base of knowledge modern technology of wiki is appropriate. Wiki is a home page which is filled with information from a group of people and can be used as a mean of accumulating knowledge on the certain topic in the process of collective work. The basis of wiki is represented by a graph, where knots are noted by keywords and vocabulary entries associated with them. Personal wiki can be integrated with other wiki's, for example in Wikipedia and other encyclopaedias.

In teaching LIS, the use of wiki is assumed for the joint performance of laboratory exercises and course projects, and also for group discussions having a possibility to give references on additional materials. Upon completion of every block of the course themes, students apply the acquired knowledge in practice and by wiki resources to collect new ideas, descriptions of interesting decisions etc. relating to this block of themes. In the future, they can be taken into account in a new modification of the course or to create independent educational content themselves.

To provide the course with a great number of aggregated information sources, it is possible to use podcasts. Podcasts are programs of subscription on a receipt of digital audio or video recordings, which can be delivered to personal computers, mobile telephones or MP3/MP4-players.

Podcasts will be used in the teaching of LIS for distribution among students by audio and video recordings including courses or comments on a study programme. Creation of podcasts is also planned by students themselves, summarising the results of their research in LIS course.

6 Conclusion

Recent developments in information technologies and telecommunications facilitate the development of new training and educational methods and tools, as described above. This provides possibilities for organising educational processes not only in the traditional way, but also by means of active learning, combining IT technologies with modern pedagogical approaches. This is of special importance for LIS teaching, where IT is the main subject of the course. The presented concept of Web 2.0 based e-teaching environment opens new horizons active teaching, providing student with wider education possibilities in enhancing their professional skills and abilities. By using Web 2.0-based LIS e-learning environment, both lecturers and students are able to create individual centres of teaching and researches on the different themes of LIS. Moreover, the designed e-environment will provide opportunities to form student personal portfolio achievements in studies and research, by submitting them in an electronic form for discussion and debriefing with co- students and lecturers.

References

1. Merkuryev, Y., Tolujev, J., Blumel, E., Novitsky, L., Ginters, E., Viktorova, E., Merkuryeva, G., Pronins, J.: A modelling and simulation methodology for managing the Riga Harbour Container Terminal. SIMULATION 71(2), 84–95 (1998)
2. Ginters, E. (ed.): Logistics Information Systems, Part I, II. Jumi Ltd., Riga (2002)
3. Soshko, O., Merkuryev, Y., Merkuryeva, G., Bikovska, J.: Development of active training and educational methods in logistics. In: Annual Proceedings of Vidzeme University College. ICTE in Regional Development, Vidzeme University College, pp. 62–66 (2005)
4. Merkuryev, Y., Merkuryeva, G.: Education in logistics – experiences and further development. In: TransBaltica 2002 Conference Materials, RMS, pp. 137–142 (2002)
5. Rikure, T., Novickis, L.: Quality Evaluation methodologies for e-Learning systems (in frame of the EC Project UNITE). EC Project IST4Balt News Journal 2 (2006)
6. Anohina, A., Grundspenkis, J.: Learner's Support in the Concept Map Based Knowledge Assessment System. In: Proceedings of the 7th European Conference on e-Learning, Agia Napa, Cyprus, November 6-7, pp. 38–45 (2008)
7. O'Reilly, T.: What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. O'Reilly Media, Inc., Sebastopol (2005)
8. How businesses are using Web 2.0: A McKinsey Global Survey. McKinsey Quart. (2007)
9. Young, G.: Technology Product Management & Marketing Professionals. Forrester (2007)

Building a Learner Psychophysiological Model Based Adaptive e-Learning System: A General Framework and Its Implementation

Tatiana Rikure and Leonids Novickis

Institute of Applied Computer Systems, Riga Technical University, Meza str. 1/3, LV-1049, Riga, Latvia
{rikure,leonids.novickis}@cs.rtu.lv

Abstract. The capability of recognizing the „human factor" considerably improves the Human-Computer-Interaction process and the impact of learning as well. High efficiency of a learner psychophysiological model based e-Learning systems is achieved due to adaptation ability to learners' real-time emotional behavior during training session. In the paper an approach for building adaptive Learning systems with a model of learner's psychophysiological state is discussed. Biofeedback sensors are used to get real-time data about user's psychophysiological state during training sessions. The research results on measuring and analyzing user's psychophysiological responses from biofeedback sensors are described. Idea of "dual adaptation" is presented. Case study of the conducted by author research experiments is presented.

Keywords: Learners' modeling, Psychophysiological state, Learning system, Adaptation, Biofeedback sensors.

1 Introduction

E-Learning is a term most frequently used for web-based and distance education. However, much broader definition may include all types of technology-enhanced learning, where technology is used to support learning process [17].

In adaptive e-Learning systems *User Model* contains information about every learner and is used by the e-Learning system for adaptation purposes [1, 14]. Therefore model of a learner's psychophysiological state allows considering learner's emotional and physiological states during technology-based learning.

Influence of the emotions on the learning process has been widely discussed and studied recently [3, 5, 6, 12, 15]. *Affective computing* investigates methods for enabling computers to recognize, model, understand, express and respond to human emotions effectively [4].

2 Learners' Affective State

For the recognition of the affective state of the learner the following main tasks should be performed: measuring or tracking the learner behavior and further interpretation of the gained data.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 31–38, 2010.

Different characteristics of the learner could be obtained during the learning process, such as, voice, facial expressions, gestures and body movements, physiological data (heart rate, blood pressure, conductivity of the skin, etc.), other human-computer interaction features.

There are technologies and methods available to perform the measurement phase: digital cameras and image processing, eye-tracking, sound processing, biofeedback sensors, as well as specially developed devices for learners' psychophysiological state monitoring. The challenging issue in this context is the interpretation of the gained data in order to recognize the learner's affective state.

Emotions can be expressed in many ways with varying intensity by different people. Therefore affective state recognition usually is modeled as a pattern recognition or fuzzy classification task.

Since the spectrum of emotions is very diverse and the expression of emotions differs, it is almost impossible to have a single model for accurate emotion prediction. The current solutions in the field use different models to infer user's emotions or create their own models based on experimental data [3, 5, 6, 7].

The ability to recognize the affective state of a learner is used in existing learning systems mainly in order to provide affective communication and system's adaptation [3, 6, 8, 9, 10, 11, 12].

First of all, it is used for the development of empathic educational agents, which are able to display an appropriate emotional response to the learner in a form of an animated character either generate emotional content of the system dialog.

Another common application scenario is adaptation of the learning system to the current affective state of the learner. Learning system's adaptive reaction usually means adaptation of the content planning and sequencing, tutoring strategies and dialogues, and other personalization issues.

However, the adaptation of the learner itself (i.e. learner's psychophysiological state during training) usually is not considered.

3 Case Study: Building Adaptive e-Learning System

3.1 Detecting Psychophysiological State of a Learner

Human mental and physical states are closely interrelated. Changes in human affective state are directly accompanied by appropriate physiological responses, such as changes in heart rate, respiration, galvanic skin response etc. Methods of physiological monitoring have been proven to be a sensitive measure for describing human emotional and physiological states in different application areas [3, 4, 13].

Therefore a method of learner's psychophysiological state monitoring with biofeedback sensors was selected and implemented in the frame of the presented research. A general recommendation for the recording of learner's psychophysiological data is to record several physiological markers simultaneously [1]. Using such multisensorial approach increases the reliability of the measurements. The selection of physiological markers to be measured depends on a purposes and characteristics of an e-Learning system.

During the research experiments the following physiological markers for the estimation of the learner's psychophysiological state were measured [2]:

- pulse (heart rate);
- systolic arterial pressure;
- galvanic skin response;
- chest breathing (respiration) frequency;
- diaphragm breathing depth;
- blood filling of the microcapillaries (using photoplethysmography method);
- physical activity (i.e. user's body movements).

During training sessions learners were equipped with a set of biofeedback sensors. For the technical implementation of the biofeedback sensors' control blocks programmable microcontrollers were used. The physiological data from the sensors was wirelessly transmitted to the receiver blocks of the psychophysiological data recording unit (Figure 1.).

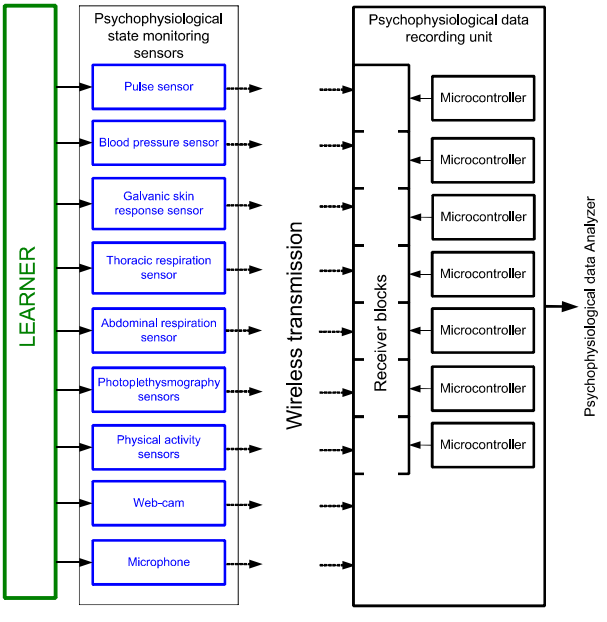


Fig. 1. Monitoring psychophysiological state of a learner

Conducted research experiments approved the reliability and appropriateness of using the selected physiological markers for detecting psychophysiological state of a learner during the training session.

Although physiological parameters certainly add important information to the student model, the psychophysiological monitoring is often referred to be expensive and obtrusive for using in learning systems. Apparently the development of modern wireless, unobtrusive micro- and nano- sensor technologies would facilitate its usage for the purposes of learner's monitoring.

It is obvious that in order to have a minimal disturbance of a learner, a minimal number of intrusive sensors should be attached. It is preferable that sensors can be easy self-attached by the user. Furthermore, intrusive sensors should be as small as possible, with high sensitivity and good reliability.

In the presented approach a model of a learner psychophysiological state is built based on the obtained psychophysiological parameters of the individual user, which are analyzed and processed using specially developed algorithms for the purposes of the conducted research.

3.2 Controlling Psychophysiological State of a Learner

Several research experiments have been organized recently by author for the purposes of studying the psychophysiological state of the learner [1, 2]. In the frame of the conducted research the hypothesis was proposed regarding the necessity to control the learner psychophysiological state during technology-based learning in order to achieve and maintain the best or optimal state for learning.

Current psychophysiological state of a human influences the efficiency of its performance, both physical and mental [4, 13]. The relatively high performance level is achieved during active phase of human wakefulness state. Furthermore, after achieving maximal level of performance efficiency, emotional intensity is increasing greatly and current psychophysiological state is followed by overwrought and fatigue with fast decrease of performance efficiency (Figure 2.).

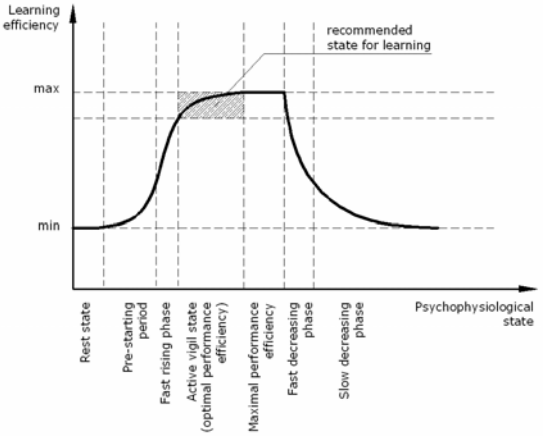


Fig. 2. Performance efficiency of a learner

Therefore the best state for learning (with relatively high levels of efficiency and stability) is the last phase before achieving maximum level. In the presented research it is referred as an *optimal (or recommended) psychophysiological state* for perceiving educational information.

Result analyses of psychophysiological research showed that it is possible to carry out control and management of human psychophysiological state using external

influences on human sense organs and perception channels. Based on the results of this research it was determined that the strongest efficiency of impact can be reached when implementing complex influence on several sense organs of the learner. For reaching aims of this research work usage of different effectors is offered, that will realize this complex external impact on the learner with the aim to keep and maintain the recommended psychophysiological state for training and learning.

3.3 Dual Adaptation

Presence of learner's psychophysiological state model in computer-based training system does not guarantee optimum choice of the next training information portion. It is because there is a great possibility that the next educational portion will not be acquired because of the non-adequate psychophysiological state of the learner. It happens because educational information is acquired much slower when the psychophysiological state of the learner does not match the active vigil state – considered to be recommended psychophysiological state for teaching and learning. The time during which the learner returns to the state applicable for training could be quite long and can exceed time resource allocated for training. This state significantly limits algorithm possibilities used in many computer-based tutoring systems. Usage of them will show good results only with the pre-condition that the learner is in recommended for training psychophysiological state.

Thus the most actual challenge becomes - facilitation of knowledge acquisition process, increasing human ability to perceive and process the incoming information flow. That is why in this research work it is offered to adopt the learner itself through control and management of his psychophysiological state during training process; with the aim to maintain it on the level necessary for training where optimum perception speed for educational material perception is achieved.

Therefore in the proposed approach the idea of *dual adaptation* means simultaneous organization of two types of adaptation during technology-based learning: adaptation of the system to the user's individual characteristics on one side and adaptation of the user's psychophysiological state for more effective perception of the learning content on the other side.

3.4 Learner's Psychophysiological State Controlling Module

For maintenance of computer-based training process using new approach with dual adaptation in this research work it is offered to implement developed model, methods and algorithms in the way of special *module* which extends traditional computer-based training systems.

Learner's psychophysiological state controlling module of an adaptive Learning system in the frame of the presented approach includes the following core features (Figure 3):

- Monitoring of a learner's psychophysiological state (biofeedback sensors, psychophysiological signal processing and analysis);
- Effecting a learner's psychophysiological state (actuating mechanisms or effectors and theirs controlling units);
- Model of learner's psychophysiological state.

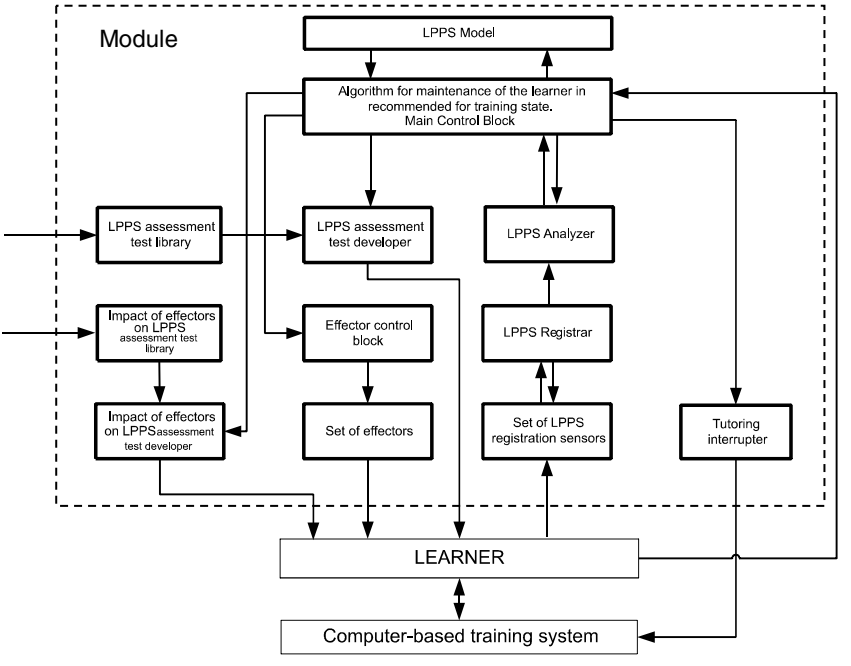


Fig. 3. Learner's psychophysiological state monitoring and control module[1]

Above mentioned module operates in two modes: test and run modes. During the "test mode" a model of learner's psychophysiological state is build. While "run mode" is a normal operating mode during training sessions in adaptive Learning system . Figure 3 shows autonomous realization of proposed module.

3.5 Building Web-Based Adaptive Learning Environment

The main advantage of autonomous realization approach is possibility to use the developed module in any existent tutoring system without its rebuilding.

The typical structure of web-based adaptive learning environment includes the following components:

- learner model
- adaptive methods
- intelligent interface
- didactic model
- domain model..

Existing adaptive hypermedia systems support different adaptive methods [18]: adaptive presentation of the text, direct quidance, adaptive sorting, hiding of links, annotation of links, adaptive multimedia presentation and map adaptation.

[1] LPPS – Learner's Psycho-Physiological State.

Adaptation in educational web-based environment usually is based on the level of a learner"s knowledge. To describe a learner"s knowledge level, an overlay model based on the application domain model is most commonly used [19]. The domain model is represented as a network of domain concepts. The network of concepts is implemented in hypermedia with the help of pages linked by references. Every page holds some information on the concept and one or several references to the relevant pages. The overlay model of a learner"s knowledge is represented as a set of pairs "concept – value" for every domain concept and serves to implement adaptation technologies. Integration of learner"s psychophysiological state controlling module into e-learning environment allows to support adaptive learning process in more flexible way. That means the approach allows improving efficiency of the existent computer-based tutoring systems getting not only positive didactic, but also essential economic effect.

4 Conclusions

Adaptive systems development is a challenge intended to improve the efficiency of learning systems. This paper describes the research conducted by author in the frame of building adaptive learning systems with a model of a learner's psychophysiological state.

New generation computer-based tutoring systems should be able to follow and control physiological and emotional state of the learner. Such interdisciplinary approach towards development of adaptive e-learning systems will let significantly improve quality of human-computer-interaction as well as efficiency of computer-based training and learning.

Further research directions include evolution of the proposed computer-based training with dual adaptation model and investigation on its adaptation possibilities for other computer-based training systems.

References

1. Rikure, T., Novitsky, L.: Studying of the Learner Psychophysiological Behavior in Human-Computer Interaction. In: Annual Proceedings of Vidzeme University College, ICTE in Regional Development, pp. 83–87. Vidzeme University College, Latvia (2007)
2. Rikure, T., Novitsky, L.: Psychophysiological Signal Processing for Building a User Model in Adaptive e-Learning Systems. In: Proceedings of the 4th WSEAS International Conference on Cellular and Molecular Biology, Biophysics and Bioengineering BIO 2008, pp. 122–125. WSEAS, Spain (2008)
3. Reichardt, D., Levi, P., Meyer, J.-J.C. (eds.): Proceedings of the 1st Workshop on Emotion and Computing - Current Research and Future Impact. University of Bremen, Germany (2006)
4. Picard, R.W.: Affective Computing. MIT Press, Cambridge (1997)
5. Lisetti, C.L., Nasoz, F.: Using noninvasive wearable computers to recognize human emotions from physiological signals. EURASIP Journal on Applied Signal Processing (11), 1672–1687 (2004)

6. D'Mello, S.K., Craig, S.D., Gholson, B., et al.: Integrating Affect Sensors in an Intelligent Tutoring System. In: Affective Interactions: The Computer in the Affective Loop Workshop at 2005 International Conference on Intelligent User Interfaces, pp. 7–13 (2005)
7. McQuiggan, S.W., Lee, S., Lester, J.C.: Predicting User Physiological Response for Interactive Environments: An Inductive Approach. In: Proceedings of the 2nd Artificial Intelligence for Interactive Digital Entertainment Conference, pp. 60–65 (2006)
8. Sarrafzadeh, A., Alexander, S., Dadgostar, F., et al.: How do you know that I don't understand? A look at the future of intelligent tutoring systems. In: Computers in Human Behavior, pp. 1342–1363 (2007)
9. Nkambou, R.: Towards Affective Intelligent Tutoring System. In: Proceedings of the Workshop on Motivational and Affective Issues in ITS, 8th International Conference on ITS 2006, pp. 5–12 (2006)
10. Wang, H., Chignell, M., Ishizuka, M.: Improving the Usability and Effectiveness of Online Learning: How can Avatars help? In: Human Factors and Ergonomics Society Annual Meeting Proceedings, pp. 769–773 (2007)
11. Beckstein, C., Denzler, J., Fothe, M., et al.: A Reactive Architecture for Ambient e-Learning. In: Proceedings of the Workshop Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments, pp. 1–10 (2007)
12. Gütl, C., et al.: AdeLE (Adaptive e-Learning with Eye-Tracking): Theoretical Background, System Architecture and Application Scenarios. European Journal of Open, Distance and e-Learning, 1–16 (2005)
13. Александров Ю.И.: Психофизиология. Питер, Санкт-Петербург (2004)
14. Nguyen, L., Do, P.: Learner Model in Adaptive Learning. Proceedings of World Academy of Science, Engineering and Technology 35, 396–401 (2008)
15. O'Regan, K.: Emotion and E-Learning. Journal for Asynchronous Learning Networks, Sloan-c 7(3), 78–92 (2003)
16. Rikure, T., Novickis, L.: Evaluation of the e-Learning Applications and Systems. In: Scientific Proceedings of Riga Technical University, Computer Science, Applied Computer Systems, pp. 104–112. RTU, Riga (2007)
17. Wikipedia-the free encyclopedia, `http://en.wikipedia.org/wiki/E-learning`
18. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. User Modelling and User – Adapted Interaction 6(2-3), 87–129 (2005)
19. Galeev, I.: Architecture of Integrated Learning Environment. In: Access to Knowledge: New Information Technologies and the Emergence of Virtual University, pp. 167–206. Elsevier Science and International Association of Universities (2003)

Quality of Study Programs: An Ecosystems Perspective

Marite Kirikova, Renate Strazdina, Ilze Andersone, and Uldis Sukovskis

Riga Technical University, Latvia
marite.kirikova@cs.rtu.lv, renate.strazdina@lv.ey.com,
ilze.andersone@rtu.lv, uldis.sukovskis@rtu.lv

The quality of study programs is one of the issues that are essential in the turbulent global environment universities nowadays operate in. Quality may be considered in terms of different quality standards trying to follow their guidelines formally and practically. This paper takes a different view of the quality issue with respect to the study programs in the field of engineering, namely, the quality of the study program is considered from the ecosystem perspective and value exchange between different members of the ecosystem is taken as a central object of interest in defining and supporting the high quality of the program. While analysis of value exchange and detection of changes in the value provision and request are not a natural part of the university teaching process, appropriate models and support systems can help to understand the value exchange process in the educational ecosystem. The understanding of the value exchange process, in turn, helps to identify and monitor knowledge requirements for developing high quality study programs.

1 Introduction

Engineering considerably differs from other fields of education with its need to provide technical skills together with a deep understanding of natural and socio-technical phenomena. All subjects taught at a university can be divided into three knowledge groups - *basic*, *field specific theoretical* and *field specific technical* knowledge. Each knowledge group has a specific frequency of changes; the basic knowledge changes slowly, field specific theoretical – faster, but the field specific technical knowledge changes with the speed of industrial innovations, thus having the highest frequency of changes among the above-mentioned knowledge groups. This classification of knowledge is similar to the one introduced by Zack, 1999 with respect to business knowledge [1]: core, advanced and innovative knowledge. In this classification, with time, advanced knowledge tends to become core knowledge, and the same also applies to the innovative knowledge, which first moves to the category of advanced knowledge and then tends to become core knowledge. Some transition of knowledge happens also among the engineering knowledge groups, however one can observe the difference, e.g., out-dated technical knowledge usually is not a part of up-to-date field specific theoretical knowledge. In order to make transparent the different types and flows of knowledge relevant in engineering education we propose using knowledge requirements monitoring systems that are based on the engineering education ecosystems model. The use of ecosystems paradigm gives an opportunity to

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 39–46, 2010.

utilize simultaneously two relevant systems approaches for knowledge requirements analysis: (1) value networks and (2) feedback mechanisms. Those two approaches allow detecting essential points for knowledge requirements fusion and support knowledge requirements acquisition and analysis with information systems solutions of different levels of complexity including the use of agent technologies.

The research work presented in this paper is a part of larger investigations into the area of educational ecosystems. A blueprint of multi-fractal knowledge management system engineering education was discussed in [2], feedbacks in the ecosystem's subsystem "School-University-Industry" were analyzed in [3]. Information fusion issues and solutions in subsystem "School-University" concerning knowledge requirements are described in [4] and [5]. In this paper we focus on the knowledge requirements in the "University-Industry" subsystem by analyzing knowledge requirements sources and value and information flows between the university and knowledge requirements sources identified in the "University-Industry" subsystem of the educational ecosystem.

The paper is structured as follows: Related works are discussed in Section 2. Section 3 presents the model of information systems support for knowledge requirements identification and monitoring in the "University-Industry" sub-ecosystem. A potential impact of the use of knowledge requirements identification and monitoring system on quality of educational programs, the solution validation results using value network theory, and directions of future work are discussed in Section 4. Brief conclusions are presented in Section 5.

2 Related Works

There are different approaches and standards used for achieving a high quality of study programs [6]. Most of them focus on internal procedures in university departments and only a few require a profound analysis of industrial requirements with respect to university graduates [7]. Usually the analysis of customer satisfaction is understood as filling out questionnaires and performing a statistical analysis of this type of information acquisition results. In our research we address the industry's needs differently, - using the ecosystems approach, i.e., considering the university and industry relationship from the point of view of mutual benefit and benefit for the student. Another wave for ensuring the quality of study programs is conceptual modeling of study content [8]. However, the essential problem is that the way knowledge is represented in university context considerably differs from the way it is represented in the industrial context [9]. Our intention is to develop an information systems solution that could support the ecosystems view and fuse different information representations in the Industry-University collaboration context.

2.1 Educational Ecosystem

The ecosystems approach has recently become popular in educational context [2], [10]. Industry, Science and School emerge as main collaborators of the university when focusing on university study programs [2], [3]. Different forth and feedback relationships may be identified among these collaborators [3]. Knowledge flow

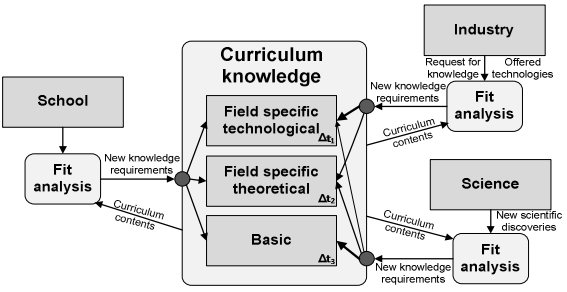


Fig. 1. Knowledge requirements sources: an ecosystems approach. $\Delta t_1 < \Delta t_2 < \Delta t_3$.

related relationships are the most essential for achieving a high quality of study programs. Therefore the fit between different knowledge contents and expectations of ecosystem's members is to be analyzed in order to achieve a transparent picture of knowledge correspondences (Fig.1)

Figure 1 shows how ecosystems approach applies to the research discussed in this paper. Knowledge provided in schools, scientific discoveries and industrial knowledge - all are to be taken into consideration when developing study programs. The courses in the study program may be grouped in three mutually related groups, namely: basic, field specific theoretical, and field specific technological. Each group of courses has a different intensity of the impact from the ecosystem's members (shown by thickness of the arrow). On the other hand, we assume that the frequency of changes in basic courses is low; the quickest changes are in technology courses, and field specific theoretical knowledge changes slower than technological one and faster than basic knowledge to be provided to students. Thus the industry appears to be one of the essential change drivers in the university education, and further in the paper the ecosystems approach will be applied for analysis of the University-Industry ecosystem abstracting from the other members reflected in Figure 1.

2.2 Models of University-Industry Cooperation

There are different universities and industry collaboration models in existence. One way of classifying the models is to divide them by area of main applied activity. (1) *Education process* (regular and vocational education) – these types of the models are the following: Collaboration in development of the curriculum, participation of business people in the education process, providing internship placements for students, and providing extra education for employees of the industry representatives. (2) *Research* – these types of the models are the following: Conducting joint scientific projects with businesses, R&D activities, collaborative research, contract research, technology transfer, and joint publications. (3) *Organizational activities* - these types of the models are the following: Joint conferences, mutual visits, joint participation in exhibitions/fairs, industrial support to student associations, and industrial representation on governing boards of higher education establishments.

Every collaboration model is a set of elements (university, industry representative, government) and relationships among these elements. The collaboration model of

"Education process" is the most important in the context of this paper. The problem identified during the analysis of the relationships is the following: How to detect the information/knowledge flows timely in order to adapt the curriculum. In order to solve the problem the information systems solutions for knowledge requirements identification and monitoring are analyzed in the research and the internal model of "University-Industry" sub-ecosystem is developed.

2.3 Knowledge Requirements Modeling and Fusion

While there are different ways of representing knowledge, communication of its contents is usually done by using some knowledge mapping technique [11], i.e. a kind of map is developed as an abstract indicator of knowledge contents. Different industrial companies and universities use various mapping techniques (if any), e.g., knowledge taxonomies, ontologies, etc. [8], [12] therefore one of the problems in an ecosystems context is the establishment of a particular knowledge model that can be shared by all the ecosystem's members [13]. Different knowledge mappings and free text descriptions are to be merged into such a knowledge model. We use a gradual, study program oriented multilevel information fusion approach to establish such a model. Information fusion is the process of utilizing one or more data sources over time to assemble a representation of aspects of interest in an environment [14].

In the framework of basic components of information fusion we use multiple knowledge identification and acquisition services and knowledge fit analysis services, situation assessment is the task performed with the use of a curriculum meta-model maintenance service.

3 Information Systems Solution

In order to identify, monitor, reflect and anticipate changes in knowledge requirements for both university and industry representatives and to support the curriculum development and amendment processes we propose to develop a supporting education-industrial information system (EIIS).

The architecture of an intended EIIS therefore has to address problems by providing multiple ways of information gathering, fusion and representation. The part of EIIS architecture that addresses these problems is presented in Figure 2. The architecture of this solution consists of four layers. The first layer consists of a number of information acquisition and analysis services that serve as first level information fusion points. Each of these services is tuned for a particular type of source of knowledge requirements. Depending of the level of intelligence the service may offer the following support for information fusion:

- Knowledge requirements source change identification
- Knowledge requirements source change representation by keywords visualization
- Knowledge requirements change analysis using a particular artificial intelligence technique

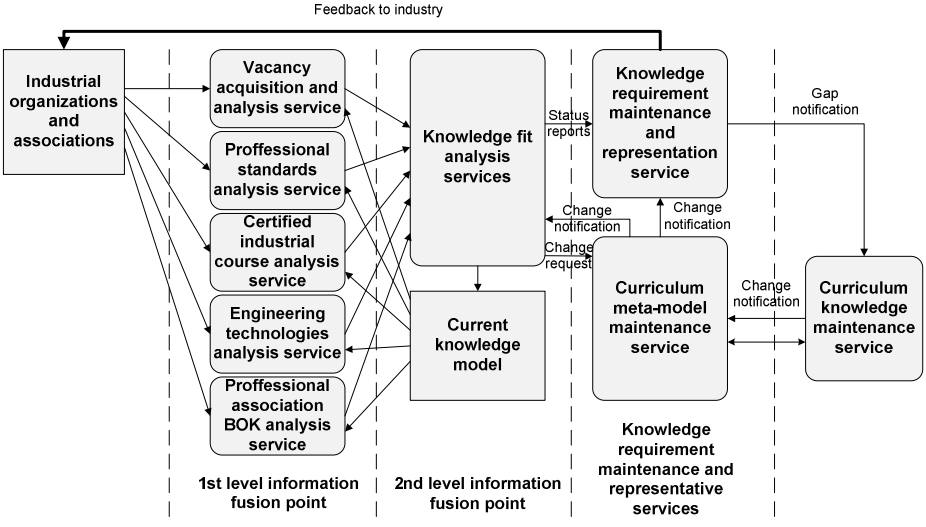


Fig. 2. A subsystem of EIIS

The second layer represents the second level information fusion points and consists of knowledge fit analysis services. In case of several study programs, each program has its own service. This service maintains a knowledge model that is an extension of the meta-model of a particular curriculum (study program). This model is developed by a gradual two side decomposition (knowledge from information acquisition services and curriculum meta-model) up to the level where meaningful information fusion operation is possible.

At the third level two mutually related services operate. The curriculum meta-model maintenance service helps to adjust the curriculum to industrial needs. The knowledge requirements monitoring and representation service calculates and represents different statistics about the fit between the study programs and industrial knowledge requirements, such as vacancy statistics, student interests profiles, certified course knowledge coverage, etc. These services are related to the curriculum knowledge maintenance service situated at the fourth level of EIIS.

Currently a change notification agent for vacancy analysis service is developed and tested [9], and some algorithms for knowledge fusion are developed. For example, with respect to the field specific knowledge the following keyword visualization supported algorithm is used.

Original knowledge developed by field experts is presented in Figure 3a. It includes several theoretical field specific knowledge requirements (Operating systems, Programming languages, Graphics etc.) and technological knowledge requirements (C++, DirectX, etc.). Each technological requirement is related to a particular theoretical requirement. Suppose that a particular employer puts forward the following requirements: Experience in C/C++ programming, Linux knowledge, Experience with graphical tools (SDL, DirectX, OpenGL, etc.), PHP knowledge and experience with SQL. Keyword identification sub-service identifies and marks some of the

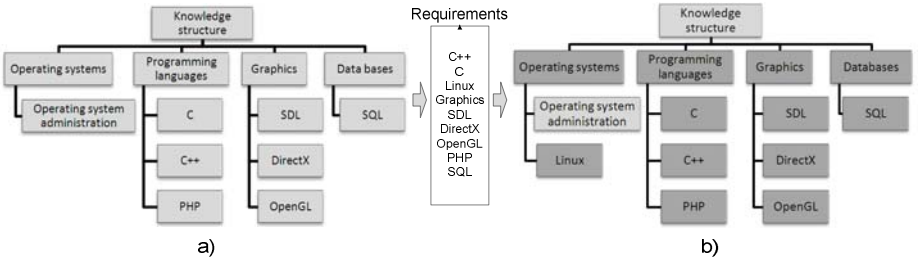


Fig. 3. Example of field specific knowledge structure (theoretical knowledge represented by the first level of hierarchy, technological – by lower level of hierarchy): a) original structure and b) highlighted requirements of employer in extended knowledge structure

requirements (these are underlined) but not all of them. Linux is definitely a knowledge requirement yet it is not marked. It happens because this requirement is not included in the original knowledge structure. Therefore knowledge representation structure has to be extended with the new knowledge requirement.

Figure 3b depicts extended knowledge structure which will be used in future keyword search. Requirements of the employer are highlighted. Although high level knowledge requirements such as Operating systems or Graphics were not directly mentioned in the job advertisement, they are nevertheless highlighted as there exist the lower level requirements related to them.

Algorithms and software for other features of the above described subsystem of EIIS are under development.

4 Assessment of the Proposed Model

The term "quality" is defined as an offering (product or service) that meets or exceeds customer requirements. In case of education there are wide discussions who are the customers of the education system – students and industry or industry alone. In different sources "customer" is defined differently, e.g. as "someone who pays for goods or services" [15] or "the person or group that is the direct beneficiary of a project or service" [16], however in both definitions it is clearly stated that the customer benefits from the "supplier". So, in this research it was assumed that the customers of the educational system are both students and industry. In order to find out whether the proposed EIIS sub-architecture helps to provide additional benefit to both types of customers, value network analysis [17] was applied.

Value network analysis is the method that investigates direct and indirect value flows among the elements of a particular system. In our case the value flows between members of "University-Industry" ecosystem are to be analyzed. The EIIS sub-architecture would be useful, if its implementation could bring some new value for ecosystem members. For value network analysis we represent the university (U) by two entities – namely, "study program" (P) and "student" (S). Then the following new value flows can be identified: (1) awareness of industry needs from I to U, (2) awareness of study program potential from U to I, (3) additional publicity from U to I, (4) profiled elective courses from P to S, (5) industry certificate oriented courses from P

to S, and other values. This shows that the proposed solution is beneficial for the "University-Industry" ecosystem in general and for the university "customers" industry and students in particular.

While from one side, the quality of the study program can be assessed by 'customers' – students and industry representatives, from the other side in the recent years governments of the member states of the European Union have undertaken the development of a common structure for European higher education and one of the primary goals is to develop more comparable quality standards. The quality standards try to cover all the requirements of the education process stakeholders. Two of available quality standards were analysed, namely, Standards and Guidelines for Quality Assurance in the European Higher Education Area [17] and QUESTE [7]. The research applies QUESTE: the Quality System of European Scientific and Technical Education-Labellisation and scoring, that is an EU initiative promoted by the European Network for Quality of Higher Engineering Education [7].

Besides other quality indicators for education and research [7] defines that quality indicators of the program design process are: '*The demands or needs of the discipline, profession, relevant industries, higher studies, and the labour market are understood by the faculty and are reflected in the program objectives, intended learning outcomes, and in the design of courses, projects, and educational activities'*. So the internal model of the "University-Industry" sub-ecosystem described in the previous section can improve the quality of the study program by providing timely and complete information about the demands or needs of the industry.

However, there are also requirements that are not addressed in the paper but are included in [7], e.g. the use of benchmarking techniques and offering continuing professional education programs, etc. So, further research should be aimed towards the extension of the model in order to address all the requirements defined in [7].

5 Conclusions

The paper shows that information systems support can enhance the cooperation between university and industry and provide value for both members of "University-Industry" ecosystem. Value network analysis reveals that particular new values flows supported by EIIS sub-architecture bring new values to main university "customers" industry and students. These value flows can help to improve study program quality and ensure their conformance to European standards, such as Standards and Guidelines for Quality Assurance in the European Higher Education Area [17] or QUESTE [7], especially with respect to understanding of needs of the profession-relevant industries and the labour market by faculty members.

References

1. Zack, M.H.: Developing a knowledge strategy. California Management Review 41(3), 125–145 (1999)
2. Kirikova, M., Grundspenkis, J., Sukovskis, U.: Educational "ecosystem" for information systems engineering. In: Horvath, I., Rusak, Z. (eds.) The Proceedings of the Seventh International Symposium on Tools and Methods of Competitive Engineering, vol. 2, pp. 769–783. Delft University of Technology, Turkey (2008)

3. Strazdina, R., Kirikova, M., Sukovskis, U.: Supporting inter-institutional knowledge feedbacks in the context of engineers' educational system. In: Chova, L.G., Belenguer, D.M., Torres, I.C. (eds.) Proceedings of International Conference of Education, Research and Innovation, pp. 912–922 (2008)
4. Strazdina, R., Stecjuka, J., Andersone, I., Kirikova, M.: Statistical analysis for supporting inter-institutional knowledge flows in the context of educational system. Accepted at the 19th International Conference on Information Systems development, Paphos, Cyprus. Springer, Heidelberg (in press) (2008)
5. Zeltmate, I., Kirikova, M., Grundspenkis, J.: The Challenges in Knowledge Representation for Analysis of Inter - Institutional Knowledge Flows. In: Sampson, K.D.G., Spector, J.M., Isaías, P., Ifenthaler, D. (eds.), Rodrigues, L., Barbosa, P. (Assoc. eds.) Proceedings of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age, Freiburg, Germany, pp. 145–152 (2008)
6. Becket, N., Brooks, M.: Evaluating quality management in university departments. Quality Assurance in Education 14(2), 123–142 (2006)
7. QUESTE: the Quality System of European Scientific and Technical Education-Labellisation and scoring, http://queste.w3sites.net/index.html
8. Subrahmanyam, G.: A Dynamic Framework for Software Engineering Education Curriculum to Reduce the Gap between the Software Organizations and Software Educational Institutions. In: Proceedings of the 22nd Conference on Software Engineering Education and Training, Washington, pp. 248–254 (2009)
9. Rudzajs, P.: Development of knowledge renewal service for the maintanace of employers' database. Bachelor thesis, Riga Technical university, Riga, Latvia (2008)
10. Uden, L., Damiani, E.: The future of E-learning: E-learning eco-system. In: Proceedings of Inaugural IEEE International Conference on Digital Ecosystems and Technologies, pp. 113–117 (2007)
11. Okada, A., Shum, S.B., Sherborne, T.: Knowledge Cartography: Software Tools and Mapping Techniques. Springer, Heidelberg (2008)
12. Guide to the Software Engineering Body of Knowledge, Institute of Electrical and Electronics Engineers (2004)
13. Chang, V., Guelt, C.: E-Learning Ecosystem (ELES) – A Holistic Approach for the Development of more Effective Learning Environment for Small-and-Medium Sized Enterprises (SMEs). In: Inaugural IEEE International Conference on Digital Ecosystems and Technologies, pp. 420–425. IEEE Press, Los Alamitos (2007)
14. Bosse, E., Roy, J., Wark, S.: Concepts, Models, and Tools for Information Fusion. Artech House (2007)
15. WORDNET, http://wordnet.princeton.edu/perl/webwn
16. TENSTEP, http://www.tenstep.com/open/miscpages/94.3Glossary.html
17. Allee, V.: Value Network Analysis and Value Conversion of Tangible and Intangible Assets. Journal of Intellectual Capital, Online version of Final Draft 9(1), 5–24 (2008)
18. Standards and Guidelines for Quality Assurance in the European Higher Education Area, Helsinki, Finland (2007)

Learning Support and Legally Ruled Collaboration in the VirtualLife Virtual World Platform*

Vytautas Čyras and Kristina Lapin

Vilnius University, Faculty of Mathematics and Informatics, Naugarduko 24, 03225 Vilnius, Lithuania
Vytautas.Cyras@mif.vu.lt, Kristina.Lapin@mif.vu.lt

Abstract. The paper addresses the purposes and design decisions produced while developing a peer-to-peer virtual world platform. The work is being done within the FP7 VirtualLife project. The purpose of the project is to create a safe, democratic and legally ruled collaboration environment. The novelty of the platform is mainly in the issues of security and trust and in the implementation of an in-world legal framework, which is real world compliant. In the paper the authors reflect on user needs and learning support in a university virtual campus, a potential scenario. The opportunities of a virtual world in enhancing learning are discussed. A new paradigm of the content is characterized as interaction versus information.

Keywords: intelligent virtual world, e-learning support, reputation management, value based interaction, virtual law.

1 Introduction

The paper presents early results obtained while developing the VirtualLife virtual world platform. It is designed under the following requirements: (1) the use of a peer-to-peer communication architecture, (2) security and trusted transactions, and (3) legally ruled collaboration. The work is being done as part of the FP7 project "Secure, Trusted and Legally Ruled Collaboration Environment in Virtual Life".

Currently VirtualLife is targeted at distance learning scenarios. The authors reflect on e-learning scenarios in 3D immersive virtual collaboration environments often called virtual worlds. Present virtual worlds are mainly leisure-based. A user may have several identities. Therefore distance education is mainly hybrid: the learning is provided in a virtual environment whereas signing a contract of a student or teacher and passing exams is performed in the real world. A trusted and secure user identity is required in order to transfer real world activities to a virtual world.

Virtual worlds offer new opportunities to enhance collaboration. Involving virtual worlds for learning provides more adequate motivation for contemporary students that cannot imagine the world without the Internet. Students perform certain activities in

* Supported by EU FP7 ICT VirtualLife project, 2008-2010, http://www.ict-virtuallife.eu

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 47–54, 2010.

the immersive 3D environment for which they obtain instant feedback. We also argue that a more elaborate legal regulation is required.

In the real world, human communities develop and enforce their internal rules whereas this is not permitted in present virtual worlds. The rules are established by virtual world creators and administrating avatars that enforce the rules. Consequently, a virtual world is like a text-based Web 1.0 platform where webpage creators influence the content. Web 2.0 enables a user to be an active creator and community builder. The user got used to be active in Web 2.0 environments and may feel restricted in a present virtual world.

2 About VirtualLife

Collaboration in a VirtualLife's virtual world is achieved through the definition of common rules that take care of all the involved cultures. A standard collection of laws and the Virtual Constitution, finalized to the creation and regulation of a secure and trusted environment (Virtual Nation), form the VirtualLife's legal framework. Hence the virtual world is not a game. The project and the software are introduced in [1].

VirtualLife architecture is based on a peer-to-peer network with nodes connected using a secure protocol. Thus the resulting virtual world is not hosted on a central server cluster but is based on a network of Virtual Zone Servers.

3 Learning Support in Virtual Worlds

This section is devoted to potential use cases and user needs. A University Virtual Campus is foreseen as a sample scenario for a validation of the VirtualLife platform. We further explore e-learning scenarios and design decisions.

3.1 Learning Needs of Today Learners

Young learners do not perceive the world without the Internet. They easily switch to a chat, e-mail and reality. This new generation is called the "digital natives" [2]. Their multitasking nature limits the learning abilities in a traditional format as traditional lessons require the lasting concentration. Traditional education requires learning and memorizing for a later use. But the students are inclined to seek for learning materials. Digital natives need an environment that supports multitasking and search.

Young people are active users of Web 2.0 applications that encourage to stay there and to return. Young people like to impress peers with curious facts. Hence social interaction and participation in group activities is their natural expectation. Platforms like Wikipedia show that their users are collaborative and altruistic contributors. Forums and blogs support communication with competent volunteers in addition to peers and teachers. MMORPG (Massively Multiplayer Online Role-Playing Game) environments have developed an instant gratification mechanism that encourages a player to achieve a higher level of skill. Learning tools based on Web 2.0 principles included the following features [3]: dynamic reward of learner's actions, visualization of learner's reputation, and a peer rating of learner's contributions.

The following ways to make learning more gratifying can be recommended [4]:

- Learning should be a combination of challenge and fun.
- The feeling of achievement should be promoted by providing instant feedback.
- Performance should be related with the status in a peer group and the grades.

Thus a modern learning environment should be learner-centered. It should support 3D visualization, context awareness, multitasking, rewarding, reputation management, contribution ratings, interactive learning objects, and chat.

3.2 Requirements of Learning Support in a Virtual World

Design decisions below are formulated considering the modern learner's needs and the experience gained using Web 2.0 based learning tools. Thus the platform becomes appealing for e-learning applications. Essential features of the developed platform such as security and trustiness would reduce the need of face-to-face meetings.

3.2.1 Encouraging the Motivation for Learning through Playful Experiences

Virtual worlds enable the creation of more elaborated and appealing learning environments comparing to text-based Web 2.0 platforms. In a 3D environment a learner immerses to a certain situation where interactive 3D objects are provided as in the PWI (Practice-World-Interaction) model [5]. In such an environment the learning of complex subjects is feasible through interaction. For example, Mantyka claims that it is hard to teach subjects involving the development of complex knowledge structures

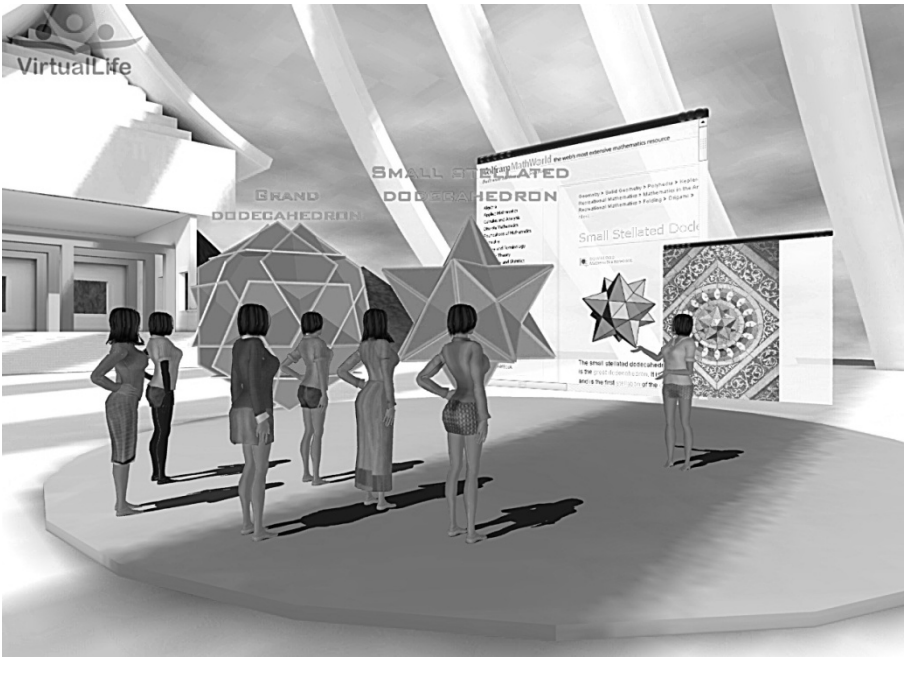


Fig. 1. Interacting with a complex spatial geometric object in a virtual world

that demand a lot of exercises, such as math [6]. But in a virtual world, an interactive object, for example, the graph of a certain mathematical function, $y = kx + b$, can be provided. The learner could observe the results and draw a conclusion while changing the coefficients k and b. Such a learning procedure accords with an active learning attitude which is expensive to introduce in the real world. Virtual worlds are more flexible and less expensive. Furthermore, students can be involved in the creation of interactive learning objects. Of course, this approach cannot cover all the topics of math but can intersperse difficult studies.

Two-dimensional environments hardly support subjects where 3D imagination is needed. For example, while teaching geometric solids in the real world environment, the teacher demonstrates paper models that help the learner to understand their composition. When teaching this subject in an e-learning environment, 3D simulations are enough. VirtualLife demo screenshot shows a sample geometry lesson where a teacher and learners interact with complex solids (see Fig. 1).

A learner gains playful experience when she interacts with learning objects and stays in a nice futuristic setting. Learning a complex material is a challenging task. But if combined with playful experiences, it makes fun and encourages to stay and repeat the actions. Fig. 2 shows an interaction with a complex geometric solid. A purpose is to understand solid's structure.

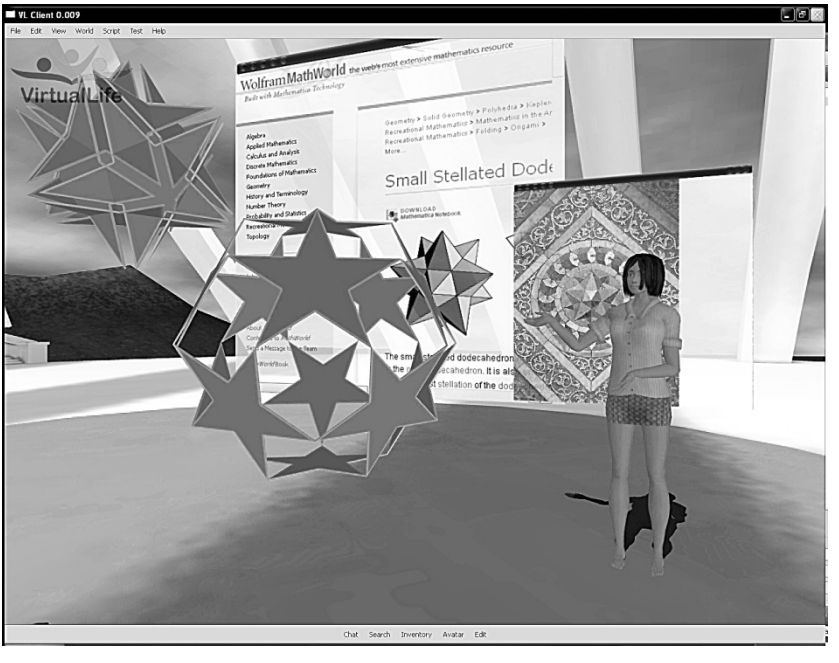


Fig. 2. Combining a complex material with a playful experience in a virtual world

3.2.2 Instant Gratification for Encouraging the Quality of Contribution

A successful learning environment has to meet the gratification challenge. It ensures user participation that is important for all online communities [7].

Instant feedback and performance is a natural feature of a synchronous communication platform. The Comtella learning environment depicts student status in the group in the form of circles which are sorted into four levels depending on their contribution along various criteria [3]. Such reputation visualization is effective when criteria are implicit. Thus it helps to avoid the gaming effect. For example, if a student knows that a certain amount of a contribution is awarded, she is encouraged to send a lot of low quality contributions.

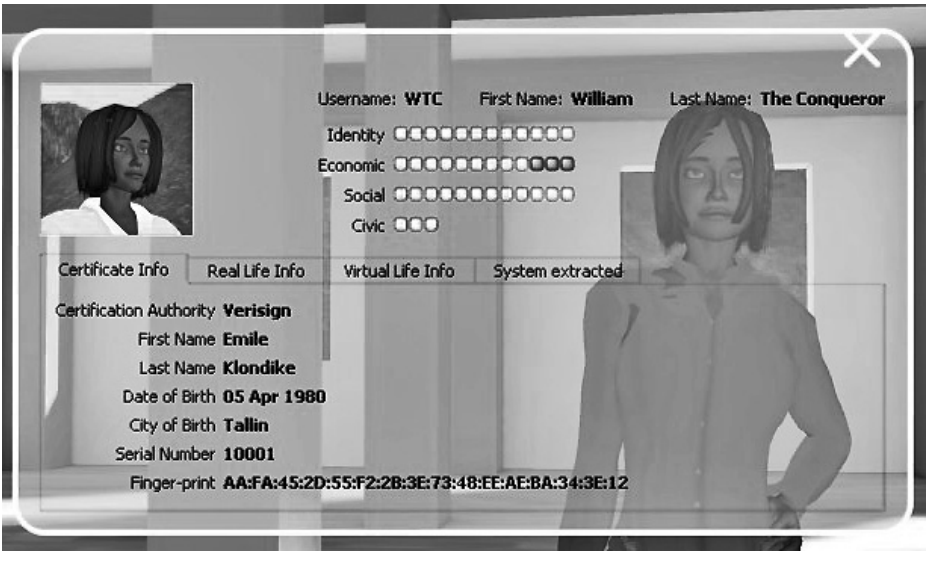


Fig. 3. Reputation visualization in the VirtualLife virtual world platform

The learner's status in a group can be achieved by reputation management. Reputation in e-business and e-learning has different meanings. Therefore VirtualLife distinguishes between civic, economical and social reputation (see Fig. 3). The civic reputation represents the avatar's status regarding the virtual law. Each violation automatically decreases it. The economical reputation is determined by avatar's behavior in economical transactions. During each transaction the avatar can rate the partner. The social reputation is determined by the users. It is rated during interactions with other avatars, e.g. evaluating the quality of learner's contributions.

In order to avoid evil-minded behavior, a negative social reputation cannot be assigned in VirtualLife. A high reputation can be gained only from positive responses for a high quality and useful contributions. This mechanism should encourage active participation and the quality of contribution. The learner is motivated to accomplish group tasks in order to achieve a high reputation in the group. An explanation of a learned knowledge to peers is also rated and is perceived as a valuable contribution.

4 Focus on VirtualLife Legal Framework

A VirtualLife legal framework was elaborated in project deliverables, see also [8]. Further the elaboration was in the form of a specification of Virtual Nation laws [9].

A virtual world is quite different from a standard video game, where there is a story, a final purpose, and the system only allows for a limited set of actions. In a virtual world there is not a determined purpose and there is not a game over. People move their avatar and establish their second life, driven by different purposes. Thus the rules of play should be replaced by a sophisticated legal framework, which is considered to be essential in order to guarantee the existence of a secure and safe virtual world. In VirtualLife, the legal system takes into account both real life values and real world laws [1].

4.1 From Norms in Law to Rules in Artifact

The legal framework is a three-tier system that is compliant with real world law. The framework is comprised of the Supreme Constitution, a Virtual Nation Constitution and sample contracts.

A Virtual Nation Constitution contains special provisions as regards, for example, the protection of objects used in that Virtual Nation under copyright law or the authentication procedure required to become a member of that nation. Distinct virtual nations, e.g. a university virtual campus and a virtual mall, are governed differently.

As one can note, the Supreme Constitution is placed in the level of contract law. This binds the user on the contractual level and contributes to law enforcement. Of course, a user of VirtualLife software is ruled not exclusively by the sources above, but also by the user's national law.

The editor of rules comprised in the VirtualLife platform is a tool to compose laws. The rule concept is approached considering Vázquez-Salceda et al. [10]. A sample toy rule 'Keep off the grass' is transformed into 'The subject – avatar – is forbidden the action – walking on the grass'. Other examples of rules, see [9]:

- An avatar is forbidden to touch objects not owned by him or a certain group.
- An avatar not belonging to a given group is forbidden to a given area of the zone.
- An avatar is forbidden to use a given dictionary of words (slang) while chatting.
- An avatar of age is forbidden to chat with avatars under age.

If an avatar violates the rule (e.g. walks on the grass), his reputation is decreased. The rules above show that the "Ought to Be reality" concept, which is used in legal theory, can be extended from the real world to a virtual world.

Rule enforcement is implemented by triggers. They trigger the changes of the virtual word states and thus invoke avatar script programs. The triggers implement a demon concept which is known in artificial intelligence. AI is an "umbrella" discipline and comprises a variety of paradigms [11].

One can note that we follow a legal informatics approach "From norms in law to rules in artifact" [12]. The approach contributes to a bridge between law and informatics. In the approach we advocate the thesis "code is law" [13].

The translation of legal rules into machine-readable format requires human intelligence. The translator faces certain problems. Just to mention a few: (a) abstractness of

norms, (b) open texture, see e.g. Hart's example of "Vehicles are forbidden in the park", (c) legal teleology, (d) legal interpretation methods (e) heuristics – to translate abstract concepts and invent low level ones.

4.2 Values Protected by VirtualLife Laws

VirtualLife laws – like laws in general – identify purposes and protected values. These are the values of a Virtual Nation. The values shall be enforced by code – a set of technologically implemented rules and laws [1].

Examples of values, which are immanent in a real-world constitution of a state, are democracy, human life, sanctity of property, legal certainty, etc. Values can be worded explicitly, but mainly they are implicit. They can be inferred from the text of a legal source and the whole legal system. For example, the Code of Conduct within the Supreme Constitution identifies equality (non-discrimination), avatars integrity, honor, reputation, privacy, free movement, freedom of thought, freedom of association, etc. Such explicit representation contributes to detect violations of the Virtual Nation laws by the users.

4.3 The Law of Avatars

The behavior of artificial agents (including avatars) shall also be governed by law – "virtual law". An example of a norm is that an avatar is forbidden to commit a (virtual) crime over another avatar. For example, an avatar is forbidden to harm (kill, hit) another avatar, steal its inventory, etc. Thus we approach a code of avatars [14].

Designers of actions over avatars have to care that such actions do not infringe the so called "virtual rights" of other avatars. For example, the physical integrity of the body of another avatar shall be preserved when my avatar takes an action – walks, moves, flies, etc. Here a question arises what the concept of an "objective right of an avatar" is. An answer can be entailed from a virtual world implementation. The right is a list of actions which are permitted to the avatar in the virtual world program.

Virtual law accords with the concept of a community of programs [15]. Many years ago Lyubimskii came to the conclusion that programs should interact similarly to humans: "the structure of the community of programs and the means of their interaction are largely similar to the structure and means of interaction in human society". Hence the interaction of programs should be ruled by similar laws.

5 Conclusions

Virtual worlds are likely to become an extension of our real lives. Therefore legal and security features have to be improved. VirtualLife implements this requirement. Certain elements of the virtual law can be implemented technologically. In a sophisticated reputation management system, avatar's activities are evaluated by the system and the users. The system detects infringements of the virtual law and decreases the civic reputation. The social reputation is influenced by teachers and learners while rating the contributions of a student.

Virtual worlds add synchronous interaction to a mere asynchronous information provision as in 2D Web applications.

Acknowledgments. Project outcomes are achieved by the consortium of 9 partners, about 30 people. The project idea originally was coined by Maria Vittoria Crispino and Francesco Zuliani from Nergal S.r.l., Italy. They push the project to the targets.

Reflections on legal informatics issues are stipulated by Friedrich Lachmayer.

References

1. Bogdanov, D., Crispino, M.V., Čyras, V., Glass, K., Lapin, K., Panebarco, M., Todesco, G.M., Zuliani, F.: VirtualLife Virtual World Platform: Peer-to-Peer, Security and Rule of Law. In: eBook Proceedings of 2009 NEM Summit Towards Future Media Internet, Saint-Malo, France, September 28-30, pp. 124–129. Eurescom GmbH (2009)
2. Small, G., Vorgan, G.: iBrain: Surviving the Technological Alteration of the Modern Mind. HarperCollins e-books, New York (2008)
3. Vassileva, J.: Toward Social Learning Environments. IEEE Transactions on Learning Technologies 1(4), 199–214 (2008)
4. Johnson, S.: Everything Bad Is Good for You. Penguin (2005)
5. Cai, H.: Service Design for 3D Virtual World Learning Applications. In: 2008 IEEE International Conference on Web Services, ICWS 2008, pp. 795–796. IEEE Computer Society, Los Alamitos (2008)
6. Mantyka, S.: The Math Plague: How to Survive School Mathematics. In: MayT Consulting Corp. St. John's, Newfoundland (2007)
7. Dodds, P.S., Muhamad, R., Watts, D.J.: An Experimental Study of Search in Global Social Networks. Science 301(5634), 827–829 (2003), http://www.sciencemag.org/cgi/content/abstract/301/5634/827
8. Spindler, G., Anton, K., Wehage, J.: Overview of the Legal Issues in Virtual Worlds. In: Proceedings of the First International Conference on User-Centric Media, UCMedia, Venice, December 9-11 (2009) (in press)
9. Cordier, G., Zuliani, F.: Virtual Nation Laws – Technical Specifications. VirtualLife deliverable (2009)
10. Vázquez-Salceda, J., Aldewereld, H., Grossi, D., Dignum, F.: From Human Regulations to Regulated Software Agents' Behavior. Artificial Intelligence and Law 16(1), 73–87 (2008)
11. Čaplinskas, A.: AI Paradigms. Journal of Intelligent Manufacturing 9(6), 493–502 (1998)
12. Čyras, V., Lachmayer, F.: Transparent Complexity by Goals. In: Wimmer, M.A., Scholl, H.J., Ferro, E. (eds.) EGOV 2008. LNCS, vol. 5184, pp. 255–266. Springer, Heidelberg (2008)
13. Lessig, L.: Code version 2.0. Basic Books, New York (2006)
14. Koster, R.A.: Declaration of the Rights of Avatars (2000), http://www.raphkoster.com/gaming/playerrights.shtml
15. Lyubimskii, E.Z.: On the Path to Building a Community of Programs. Programming and Computer Software 35(1), 2–5 (2009); translated from Programmirovanie (in Russian), http://www.gridclub.ru/library/publication.2005-11-23.7283041869/publ_file/

Rule-Based Management of Schema Changes at ETL Sources

George Papastefanatos[1], Panos Vassiliadis[2], Alkis Simitsis[3], Timos Sellis[4], and Yannis Vassiliou[1]

[1] National Technical University of Athens
{gpapas,yv}@dblab.ece.ntua.gr
[2] University of Ioannina
pvassil@cs.uoi.gr
[3] HP Labs
alkis@hp.com
[4] Institute for the Management of Information Systems
timos@imis.athena-innovation.gr

Abstract. In this paper, we visit the problem of the management of inconsistencies emerging on ETL processes as results of evolution operations occurring at their sources. We abstract Extract-Transform-Load (ETL) activities as queries and sequences of views. ETL activities and its sources are uniformly modeled as a graph that is annotated with rules for the management of evolution events. Given a change at an element of the graph, our framework detects the parts of the graph that are affected by this change and highlights the way they are tuned to respond to it. We then present the system architecture of a tool called Hecataeus that implements the main concepts of the proposed framework.

Keywords: ETL Schema Evolution, Hecataeus.

1 Introduction

In a high level description of a data warehouse general architecture, data stemming from operational sources are extracted, transformed, cleansed, and eventually stored in fact or dimension tables in the data warehouse. Once this task has been successfully completed, further aggregations of the loaded data are also computed and subsequently stored in data marts, reports, spreadsheets, and several other formats that can simply be thought of as materialized views. The task of designing and populating a data warehouse can be described as a workflow, generally known as Extract-Transform-Load (ETL) workflow, which comprises a synthesis of software modules representing extraction, cleansing, transformation, and loading routines. The whole environment is a very complicated architecture, where each module depends upon its data providers to fulfill its task. This strong flavor of inter-module dependency makes the problem of *evolution* very important in data warehouses, and especially, for their back stage ETL processes.

During the lifecycle of the warehouse it is possible that several counterparts of the ETL process may evolve. For instance, assume that a source relation's attribute is

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 55–62, 2010.

deleted or renamed. Such a change affects the entire workflow, possibly, all the way to the warehouse, along with any reports over the warehouse tables. Similarly, assume that the warehouse designer wishes to add an attribute to a source relation. Should this change be propagated to ETL activities that depend on this source? Research has extensively dealt with the problem of schema evolution, in data warehouses [1, 2, 3, 8, 11, 12] and materialized views [4, 5, 6]. Although several problems of evolution have been considered in the related literature, to the best of our knowledge, there is no global framework for the management of evolution in the described setting.

In this paper, we sketch a framework for detecting and resolving inconsistencies emerging on ETL processes as results of evolution operations. *The goal is to provide a mechanism to the designer for the smooth adaptation of ETL scenarios to evolution changes occurring at their sources as well as for the early detection of vulnerable parts in the overall design*. The proposed framework employs a representation technique that maps all the essential constructs of an ETL configuration to graphs. Thus, its basis is a graph model, called *evolution graph*, which models in a coherent and uniform way internal structural elements of an ETL process such source relations, activities, queries extracted from ETL procedures, etc.

We furthermore provide a suitable technique for handling changes occurring in the ETL source schema, in such way that the human interaction is minimized. The provided technique enriches the evolution graph with semantics, namely evolution events and rules, called policies in our framework, that predetermine the impact of changes on the graph constructs. These rules dictate the actions that are performed, when additions, deletions or modifications occur on the DW sources. Specifically, assuming that a graph construct is annotated with a policy for a particular event (e.g., a relation node is tuned to deny deletions of its attributes), the proposed framework (a) performs the identification of the affected part of the graph and, (b) if the policy is appropriate, proposes the readjustment of the graph to fit to the new semantics imposed by the change. All of the above concepts are implemented in a powerful and user friendly tool, called HECATAEUS.

Theoretical aspects concerning the employed graph model, the proposed rule-based framework as well as its experimental evaluation over real case ETL scenarios have been thoroughly presented in [9, 10]. In this paper we provide in details the internals of the system architecture of the proposed tool.

2 Graph-Based Modeling of ETL Processes

We employ a graph theoretic approach to capture the various and complex schema dependencies that exist between software modules comprising an ETL process. The proposed graph modeling uniformly covers relational tables, views, ETL activities, database constraints and SQL queries as first class citizens. All the aforementioned constructs are mapped to a graph, that we call *Evolution Graph*. The constructs that we consider are classified as *elementary*, including relations, conditions, queries and views and *composite*, including ETL activities and ETL processes. Composite elements are combinations of elementary ones. Originally, the model was introduced in [10] and here, we provide an extended summary.

Each **relation** $R(\Omega_1,\Omega_2,...,\Omega_n)$ in the database schema, either a table or a file (it can be considered as an external table), is represented as a directed graph, which comprises a *relation node*, R, representing the relation schema; n *attribute nodes*, one for

each of the attributes; and n *schema relationships*, directing from the relation node towards the attribute nodes, indicating that the attribute belongs to the relation. Constraints – i.e., primary/foreign key, unique, not null – are modeled with use of a separate *constraint node* (i.e., PK node, not null node, etc), connected via operand edges with the attribute(s) on which the constraint is applied.

The graph representation of a Select - Project - Join - Group By (SPJG) **query** involves a new node representing the query, named *query node*, and *attribute nodes* corresponding to the schema of the query. The query graph is a directed graph connecting the query node with all its schema attributes, via *schema relationships*. In order to represent the relationship between the query graph and the underlying relations, we resolve the query into its essential parts: SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY, each of which is eventually mapped to a subgraph. The edges connecting the query node with its subgraph components (i.e., attributes contained in the SELECT clause, the relation nodes contained in the FROM clause, etc.) are annotated as *map-select*, *from, where, group-by* and *having relationships.* The direction of the edges is from the query subgraph towards its source subgraphs (i.e., the respective relations/views accessed by the query). WHERE and HAVING clauses are modeled via a left-deep tree of logical conditions to represent the selection formulae; the edges involved are annotated as *operand relationships*. Nested queries are also part of this modeling, too. For the representation of aggregate queries, we employ a new node denoted as GB, to capture the set of attributes acting as the aggregators; and one node per aggregate function labeled with the name of the employed aggregate function; e.g., COUNT, SUM, MIN.

Views are considered either as queries or relations (materialized views). They constitute both queries over the database schema as far as their definition is concerned and relations to other queries as far as their functionality and their extension are concerned. Their dual role is captured and represented as intermediate graphs between relations and queries.

ETL **activity** is modeled as a sequence of SQL views. An ETL activity necessarily comprises: (a) one (or more) *input view(s)*, populating the input of the activity with data coming from another activity or a relation; (b) an *output view*, over which the following activity will be defined; and (c) a *sequence of views* defined over the input and/or previous, internal activity views.

Lastly, an ETL **summary** is a directed acyclic graph acting as a zoomed-out variant of the detailed evolution graph**.** The set of nodes comprises all activities, relations and views that participate in an ETL process and the edges connect the providers and consumers.

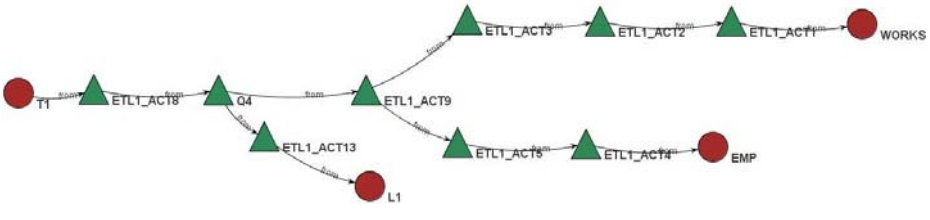


Fig. 1. Zoomed-out view of an ETL scenario

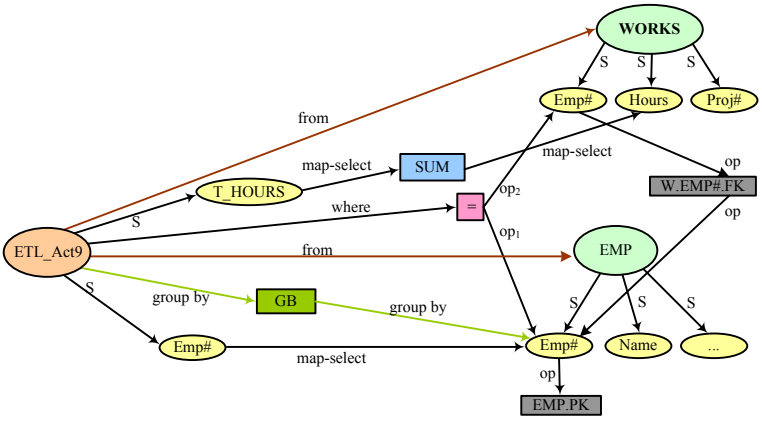


Fig. 2. Detail graph representation of ETL_Act9 activity

Figure 1 shows the summary of a simple ETL workflow involving 9 activities (green triangles) two data sources, (i.e., `EMP`, `WORKS`), one lookup table, (i.e., L_1) and one target table T_1 of the DW.

Fig. 2 depicts the detailed graph representation for a specific activity, namely `ETL_Act9` of the ETL summary, containing the following aggregate query:

```
Act9: SELECT EMP.Emp# as Emp#, Sum(WORKS.Hours) as T_Hours
      FROM   EMP, WORKS
      WHERE  EMP.Emp# = WORKS.Emp#
      GROUP  BY EMP.Emp#
```

3 Regulating Schema Evolution

The basic aspects of our framework involve the detection of the parts of the system, which are affected by an evolution change and the regulation of their reaction to this change. Therefore, we first, exploit the dependencies which are represented as edges in the evolution graph to both detect syntactical and semantic inconsistencies following a schema evolution event. We furthermore regulate the impact of an evolution event towards the nodes of the graph by *annotating the graph with rules, called policies*. The adaptation of a node to an evolution event and furthermore the propagation of the event towards the rest of the graph is dictated by the rule defined on the node. The proposed framework enables the user to proactively identify and regulate the impact of evolution processes. It provides the appropriate semantics to perform hypothetical evolution scenarios and test alternative evolution policies for a given configuration before the evolution process is applied on a production environment.

In such manner, each graph construct is enriched with policies that allow the designer to specify the behavior of the annotated construct whenever events that alter the database graph occur. The combination of an event with a policy determined by the designer/administrator triggers the execution of the appropriate action that either blocks the event, or reshapes the graph to adapt to the proposed change. The space of

potential events comprises the Cartesian product of two subspaces; specifically the space of hypothetical actions (addition/ deletion/modification) by the space of graph constructs sustaining evolution changes (e.g., nodes for relations, attributes, conditions, etc.). For each of the above events, the administrator annotates graph constructs with policies that dictate the way they will react to an event when affected. Three kinds of policies are defined: (a) *propagate* the change, meaning that the graph must be reshaped to adjust to the new semantics incurred by the event; (b) *block* the change, meaning that we want to retain the old semantics of the graph and the hypothetical event must be blocked or, at least, constrained, through some rewriting that preserves the old semantics [6, 7] and (c) *prompt* the administrator to interactively decide what will eventually happen. For the case of blocking, the specific method that can be used is orthogonal to our approach, which can be performed using any available method [6, 7].

Specifically, given an *event* altering the source database schema our framework determines those activity graph constructs that are directly connected to the source altered and thus affected by the event. For each affected construct, its prevailing policy is determined. According to the prevailing policy, the status of each construct is set. Subsequently, both the initial changes, along with the readjustment caused by the respective actions, are recursively propagated as new events to the consumers of the activity graph.

Example. Consider the simple example query `SELECT * FROM EMP` as part of the `ETL_ACT4` of Fig 1. Assume that the provider relation EMP is extended with a new attribute `PHONE`. There are two possibilities: First, the * notation signifies the request for any attribute present in the schema of relation `EMP`. In this case, the * shortcut can be treated as "return all the attributes that `EMP` has, independently of which these attributes are". Then, the query must also retrieve the new attribute `PHONE`. Alternatively, the * notation acts as a macro for the particular attributes that the relation `EMP` originally had. In this case, the addition to relation `EMP` should not be further propagated to the query.

A naïve solution to a modification of the sources; e.g., the addition of an attribute, would be that an impact prediction system must trace all queries and views that are potentially affected and ask the designer to decide upon which of them must be modified to incorporate the extra attribute. We can do better by extending the current modeling. For each element affected by the addition, we annotate its respective graph construct with the policies mentioned before. According to the policy defined on each construct the respective action is taken to correct the query.

Therefore, for the example event of an attribute addition, the policies defined on the query and the actions taken according to each policy are:

- *Propagate attribute addition*. When an attribute is added to a relation appearing in the `FROM` clause of the query, this addition should be reflected to the `SELECT` clause of the query.
- *Block attribute addition*. The query is immune to the change: an addition to the relation is ignored. In our example, the second case is assumed, i.e., the `SELECT *` clause must be rewritten to `SELECT A1,…,AN` without the newly added attribute.

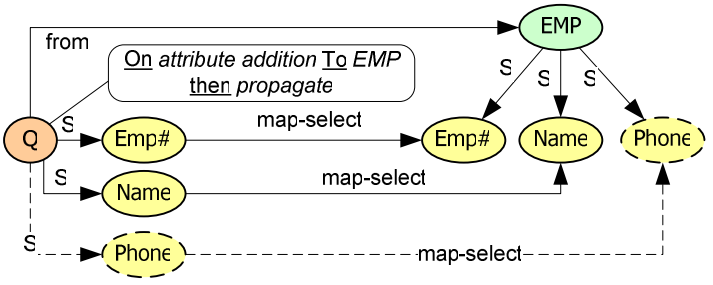


Fig. 3. Propagating addition of attribute PHONE

— *Prompt*. In this case (default, for reasons of backwards compatibility), the designer or the administrator must handle the impact of the change manually; similarly to the way that currently happens in database systems.

The graph of the query `SELECT * FROM EMP` is shown in Figure 2. The annotation of the `Q` node with *propagating addition* indicates that the addition of `PHONE` node to `EMP` relation will be propagated to the query and the new attribute is included in the `SELECT` clause of the query.

4 System Architecture

In the context of the proposed framework, we have implemented a tool, called Hecataeus, used for the construction and visualization of the evolution graph, its annotation with policies regarding evolution semantics, and lastly the management of evolution propagation towards the graph. Hecataeus enables the user to transform ETL activities abstracted as SQL source code to evolution graphs, explicitly define policies and evolution events on the graph and determine affected and adjusted graph constructs according to the proposed framework. The graph modeling of the environment has versatile utilizations: apart from the impact prediction and the creation of hypothetical evolution scenarios, the user may also assess several graph-theoretic metrics of the graph that highlight sensible regions of the graph. Hecataeus is a user-friendly visual environment that helps administrators and users to perform hypothetical evolution scenarios on database applications.

The main packages of Hecataeus are shown in the diagram of Fig. 4. The *Parser* is responsible for parsing the input files (i.e., DDL and workload definitions). The functionality of the *Catalog* is to maintain the schema of relations, views, etc., as well as to validate the syntax of the workload processed (i.e., activity definitions, queries, views) by the Parser. The *Evolution Manager* is responsible for representing the underlying schema and the parsed queries abstracted from ETL activities in the proposed graph model. The Evolution Manager holds all the semantics of nodes and edges of the aforementioned graph model, assigning nodes and edges to their respective classes. It holds all the evolution semantics for each graph construct (i.e., events, policies) and algorithms for performing evolution scenarios. The *Metric Manager* is responsible for maintaining the metrics definition and for their application on the graph. Each metric applied on the evolution graph is implemented as a separate

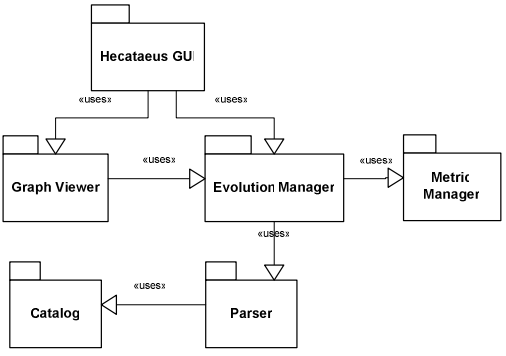


Fig. 4. System Architecture

function in the Metric Manager. The *Graph Viewer* is responsible for the management of the visual properties of the graph. It communicates with the Evolution Manager, which holds all evolution semantics and methods. Graph Viewer offers distinct colorization and shapes for each set of nodes and edges according to their types and the way they are affected by evolution events. It applies layout algorithms on the graph, adjusts the visibility of nodes and visualizes the graph at different levels of abstraction. Lastly, the Hecataeus *GUI* is responsible for the interaction with the user offering a large variety of functions, such as editing of the graph properties, addition, deletion and modification of nodes, edges and policies. The GUI package enables the user to raise evolution events, to detect affected nodes by each event and highlight appropriate transformations of the graph. Lastly, it offers the import or export of evolution scenarios to XML or image formats (i.e., jpeg).

In Fig. 5 the class diagram of the core component of Hecataeus, i.e., Evolution Manager is shown. The `EvolutionGraph` class comprises a collection of *nodes* and *edges*, which belong to a certain *type* (i.e., relation node, from edge, etc.). Each node *is annotated* with a collection of policies; each of them has a *type* (i.e., propagate, block or prompt) for *handling* an event. Additionally, a node *sustains* a collection of

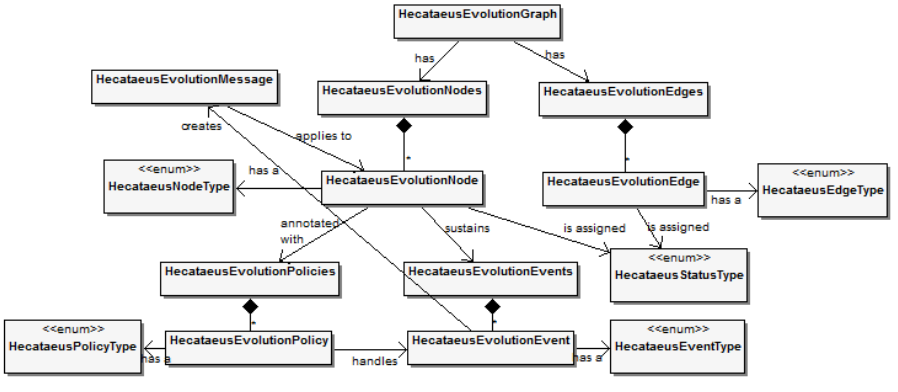


Fig. 5. Evolution Manager Class Diagram

events, which belong to a specific *event type* (i.e., delete attribute, rename relation, etc.) according to the type of node on which they occur. Lastly, a message is *created* for each event occurred on a node of the graph and transmits the impact of the event towards the adjacent nodes. Nodes handle the event and according to the prevailing policy *are assigned with* a status determining the action that is performed on them.

5 Conclusions

In this paper, we have dealt with the internals of a system, Hecataeus that handles the schema evolution in ETL environment. Our goal was to provide a coherent framework for appropriately propagating potential changes occurring at the ETL sources to all affected parts of the system, with a limited overhead imposed on both the system and the humans, who design and maintain it. Toward that aim, we have modeled the internal parts of ETL activities as the constituents of a dependency graph and we annotate parts of this graph with rules that regulate the propagation of evolution changes towards the whole workflow. In this paper we have presented the internal architecture of Hecataeus, which has been specifically designed in an extensible fashion to allow the future incorporation of different kinds of events and policies.

References

1. Golfarelli, M., Lechtenbörger, J., Rizzi, S., Vossen, G.: Schema Versioning in Data Warehouses. In: ECDM 2004, pp. 415–428 (2004)
2. Blaschka, M., Sapia, C., Höfling, G.: On Schema Evolution in Multidimensional Databases. In: Mohania, M., Tjoa, A.M. (eds.) DaWaK 1999. LNCS, vol. 1676, pp. 153–164. Springer, Heidelberg (1999)
3. Kaas, C., Pedersen, T.B., Rasmussen, B.: Schema Evolution for Stars and Snowflakes. In: ICEIS (2004)
4. Bellahsene, Z.: Schema evolution in data warehouses. Knowledge and Information Systems 4(2) (2002)
5. Mohania, M., Dong, D.: Algorithms for adapting materialized views in data warehouses. In: CODAS (1996)
6. Nica, A., Lee, A.J., Rundensteiner, E.A.: The CSV algorithm for view synchronization in evolvable large-scale information systems. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, p. 359. Springer, Heidelberg (1998)
7. Velegrakis, Y., Miller, R.J., Popa, L.: Preserving mapping consistency under schema changes. VLDB J. 13(3) (2004)
8. Bouzeghoub, M., Kedad, Z.: A Logical Model for Data Warehouse Design and Evolution. In: Kambayashi, Y., Mohania, M., Tjoa, A.M. (eds.) DaWaK 2000. LNCS, vol. 1874, p. 178. Springer, Heidelberg (2000)
9. Papastefanatos, G.: Policy Regulated Management of Schema Evolution in Database–centric Environments. PhD Thesis. NTUA (February 2009)
10. Papastefanatos, G., Vassiliadis, P., Simitsis, A., Vassiliou, Y.: What-if Analysis for Data Warehouse Evolution. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 23–33. Springer, Heidelberg (2007)
11. Wrembel, R., Bebel, B.: Metadata Management in a Multiversion Data Warehouse. J. Data Semantics (8), 118–157 (2007)
12. Golfarelli, M., Rizzi, S.: A Survey on Temporal Data Warehousing. IJDWM 5(1), 1–17 (2009)

Multiversion Spatio-temporal Telemetric Data Warehouse

Marcin Gorawski

Silesian University of Technology, Institute of Computer Science,
Akademicka 16, 44-100 Gliwice, Poland
Marcin.Gorawski@polsl.pl

Abstract. One of the crucial problems characterizing current data warehouses is the implicit assumption of dimension invariance with respect to the time dimension. This assumption inhibits the proper treatment of changes in dimension data. Meanwhile, we can give examples indicating that it is necessary to take into consideration the modifications of dimension data - ignoring such changes leads to incorrect analysis which then results in wrong decisions. This article describes the implemented temporal telemetric data warehouse system, which provides the user with the ability to query about the time interval embracing many structure versions. The system also informs the user about modifications which occurred in separated structure versions.

Keywords: data warehouse, spatial data warehouse, temporal data warehouse, structures versioning, temporal operations.

1 Introduction

Regulation of energy sector in EU created new market – media (electricity, gas, heat, and water) recipient market. The main problem is automated reading of hundreds of thousands recipients meters and critically fast analysis of terabyte sets of meters' data. Condition for achieving this goal is usage of an Integrated Meter Reading (IMR) and a Spatio-Temporal Telemetric Data Warehouse (STDW(t)) – Monitoring and Media Distribution Decision Support System (2MDSS) (fig. 1) [6, 9].

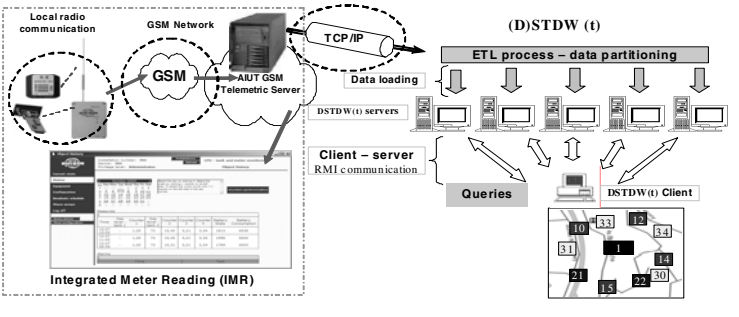


Fig. 1. The Monitoring and Media Distribution Decision Support System (2MDSS)

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 63–70, 2010.

These meters measure the usage of water (W), gas (G), and electricity (E). IMR system sends data from media meters to a database system via a cellular telephony network (GPRS). IMR is a transactional system, which services four types of meters: electrical energy, water, gas, and heat. Data stored in the 2MDSS telemetric server are in raw state and need to be adequately formatted. STDW(t) system gathers data from telemetric servers in extraction process via a network with TCP/IP protocol. In distributed STDW(t) (DSDW(t)) of 2MDSS, during extraction process additional partitioning is performed followed by data loading to system nodes [7, 8].

2 The 2MDSS Modeling

The 2MDSS modeling in a spatio-temporal scenarios is focused on classified objects {k*} and georegions {R} dimensions variability. 2MDSS objects can be classified as follows.

A. Single dimension k* objects (dimensions) – single data, data packages, and data streams, from e-receptor measurements can be presented as tuples with attributes (e.g. size, measurement frequency and period, place, name and type).
B. Spatial k* objects (spatial data types) – a spatial object can be: point (e.g. meter, counter), line or region in a 2/3D space (georegion). Spatial objects include characteristics of object geometry, metrics, relations, density, and objects decomposition.
C. Spatio-temporal k* objects (spatio-temporal data types). These objects are described with spatial data expanded with time (spatio-temporal data) and modeled with transaction or validity time. Spatio-temporal object can change its position and/or its form. The standard models have 3-5 dimensions.
D. Stream k* objects (stream data types). These objects are described with stream data, that create data sets of single spatio-temporal data and spatio-temporal data packs and data streams.

The 2MDSS works in spatio-temporal scenarios, certain Aggregates Space can include any topological georegion that has a dynamic characteristic (like in [15]).

The georegion services by 2MDSS can be denoted as the R set of two-dimensional regions (2D) with the smallest granulation of static aggregations $\{R_i\ (1<=i<=N) \mid R_i \in 2D, R_i \in R\}$ (e.g. constant number of segments if a road network or e-receptors, concentrators) or time changeable $\{R_i(t) \mid R_i(t+1) \neq R_i(t), R_i(t)/t>0\}$ (e.g. various number of e-receptors – regions of a cellular network antennas, dependable on weather conditions, capacity, etc.).

The time axis is denoted with discrete timestamps t_c $\{t_c \mid t_c \in T, 1<=c<=|T|\}$, where T is a set of ordered timestamps, describing $R_i(t)$ such that for $t_{c+1}-t_c=\Delta t>0$, the period of regions structure changes fulfills $R_i(\Delta t)>>0$ condition.

For example each counter k $\{k \mid k \in S,\ S \in R_i(t)\}$ generates one measurement – value ms_k, e.g. each $\Delta t = 15$ (minutes) in a $R_i(t)$ region, that changes (various number of counters) every 4 hours ($R_i(\Delta t) >= 240$). Each e-receptos is connected with the

ms_k value (e.g. measurements), then the $f_{agg}(1)$ function calculates number of e-receptors and the $f_{agg}(ms_k)$ calculates aggregates for qualified e-receptors.

Each of $R_i(t)$ regions is connected with the measurements aggregates set $R_i(t).f_{agg}(ms_k)$, obtained from k e-receptors localized in $R_i(t)$, which values are actualized cyclically. The $R_i(t).f_{agg}(ms_k)$ is calculated with distributive aggregation function f_{agg} (count, sum, max, min).

The 2MDSS is modeled as one of 6 characteristic STDW(t)(k*R) classes:

- STDW(t)(k*A) – constant $R_i(t)$ regions and stationary spatial objects.
- STDW(t)(k*B) – characteristic, constant $R_i(t)$ regions and stationary spatial objects – subclasses: STDW(t)(k*B1) cluster characteristic, STDW(t)(k*B2) bucket characteristic.
- STDW(t)(k*C) – slowly changing $R_i(t)$ regions and stationary spatial objects.
- STDW(t)(k*D) – quickly changing $R_i(t)$ regions and stationary spatial objects.
- STDW(t)(k*E) – constant $R_i(t)$ regions and a known set of mobile objects, which position in a t moment cannot be precisely calculated.
- STDW(t)(k*F) – constant $R_i(t)$ regions and known mobile objects trajectories and known mobile objects trajectories.

3 Multiversion STDW(t)

Previous research on *Multiversion Data Warehouse* (MVDW), concerns mainly multiversioning of classical data warehouses (DW) [2, 3, 5, 16, 18].

The majority of this researches present incremental refreshing of traditional DW forced with source data change. We can distinguish actualization of dimensions schema and a fact table. Through *dimensions schema actualization* we mean change (actualization) of a dimension structure and their instances (objects). The most important research on multiversioning of classical DW are [1, 4, 19].

The schema DW is represented as a graph with defined algebra that allows the creation of new versions [5]. The DW evolves in a direction of: a) new versions and b) upgrading schema of every previous version. The idea of history in DW is a set of versions, which includes a special type of cross-version querying. In [2, 18] the Multiversion Data Warehouse - MVDW is proposed, and is defined with a set of its versions. Each version consists of a schema version and an instance version. The two kinds of versions can be distinguished: (1) real, considering real changes in data sources and (2) alternative, considering changes for various simulation scenarios. Real versions create a linear order, while alternative versions create tree structures with data common for different versions. The concept of multiversion data warehouse is fully presented in [19].

The evolution of STDW in the 2MDSS system is presented. In a mutliversion form, through time ordered set of a data schema versions and instance versions. Our research on multiversion STDW(t) are connected with defining changes in $R_i(t)$ regions, that influence:

- Data validity time while keeping coherence and correctness of a dimensions structure change.
- Necessity of those changes (minima and complete set of the $R_i(t)$ structure).

4 The Temporal Aspects in Data Warehouses

The traditional data warehouses are ideally suited for analysis of facts changing in time [4]. It is expected that the OLAP data warehouse allows reliable data analysis even in a long time period [17]. In this context, the quite interesting fact is that data warehouses cannot efficiently manage modifications of dimensions structure (e.g. introducing the next unit or branch in a company) – even when usually in that warehouse time is represented as one of dimensions. There are many examples that show how important is the possibility to change and update dimensions in time – ignoring this fact leads to obtaining incorrect analysis results [11]. Problems like, e.g. comparison of data from different time periods or designating trends need adequate manner of managing changes in a dimensions structure. Otherwise, we have to accept the possibility of obtaining incorrect results and in effect making incorrect decisions [4,11].

The correct and consistent inclusion of dimension structures modification in time, needs describing dimensions with *time stamps,* to obtain *validity time* of a data warehouse dimensions [4].

The validity time is a time, in which "the fact is true in modeled reality" [10]. The time stamp is denoted as $[T_s, T_e]$ and means that the dimension is valid in this period where: T_s marks the period beginning, T_e marks the period end, and $T_e \geq T_s$.

When we present all time stamps of all modifications on a time axis, then the range between two time stamps on this axis marks the structure version. Through this we mean a data warehouse view that is valid in a certain time period. In one structure, the dimension structure is consistent and constant. Each modification operation for the dimension forces creation of a new structure version – actual structure loses its validity. This kind of modification operations is called *temporal operations.* In case of designing the new system such operation can be, e.g. connecting a new node in a certain point of time, connecting new meter, updating a range of operation for a node or a meter failure. We also have to introduce the idea of a *time unit* also called the *chronon.* The chronon is a time range with certain determined and undivided minimal length [10]. When setting the chronon's length simultaneously we set the precision of a data representation in a data warehouse. The length of chronon is set with consideration of the data warehouse character (data character and its usage). In case of designing the system the most important is the best and precise representation of time, because measurements data can be send to central even every couple of minutes. That is why the chronon length is set for one second.

Fig.2 presents the time axis with several data structure versions in a data warehouse. The data warehouse stores data from a certain region in which we have certain number of nodes along with meters. The user can perform analysis and queries starting from the beginning point T0. In this point, there are three nodes A, B, and C, and each of them services several meters (fig.1). Until T1 there was only one structure version, with the validity time [T0, NOW] (NOW denotes present moment). In T1, B2 meter malfunctioned, and this is the temporal operation that creates instability in structures consistency and forced the creation of a new structure version. So now there are two structure versions: SV1 [T0, T1] and SV2[T1, NOW]. Up to the "most actual" version there were three more temporal operations – in points T2, T3 and T4. So the most actual version has the SV5 identifier and its validity time is [T4, NOW]. In this time we have two nodes B and C along with meters.

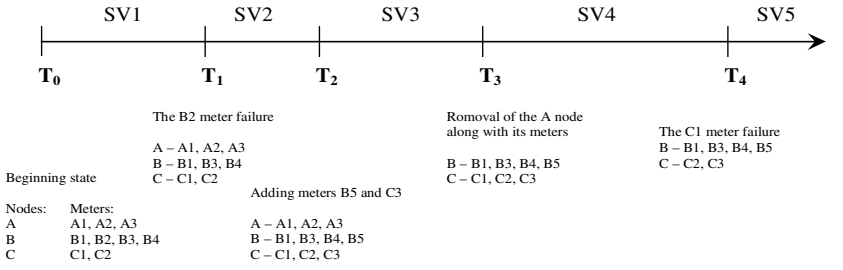


Fig. 2. An example of SDW structures versioning

5 Aggregate Tree

The designing of the aggregated structure in data warehouse systems that process spatial and temporal data is a key problem [14]. The queries performed in a spatio-temporal data warehouse system concern aggregated data so preliminary aggregation in indices greatly reduces response time. In the presented problem we implemented the so called aggregate tree, which is based on [13]. The aggregate tree is an index created by data warehouse systems in the operation memory. This approach considerably decreases query response time (in comparison to performing query directly in a base system). Before creating the aggregate tree certain factors should be defined, e.g. tree height and a size of a net created from so called minimal bounding rectangle – MBR. The MBR is the smallest, indivisible fragment of space (map), for which we aggregate data stored in the database. If there is a need to collect aggregated results from an area smaller than MBR, then we have to increase the MBRs grid density. Along with the increase of the tree height and the MBRs number (which is consistent with the increase of a MBRs grid density overlaid on the map) the tree construction time also increases and the ability to perform more adequate queries emerges. The user will have the ability to modify the tree parameters, so he can choose them empirically, for to compromise query precision and a tree creation time.

6 Data Modeling

The schema of STDW(t) based on the so-called cascaded star [9, 12] is shown in fig.3. For best understanding the schema includes only tables along with connecting relations. On the highest abstract level the schema consists of a main fact table – INSTALLATION and five dimensions: NODES, METERS, WEATHER, MEASURES, and MAP. These dimensions store data about nodes and meters along with their attributes, weather conditions, measures from the meters, and the terrain map. The separate sub dimension tables of the main dimensions, store attributes connected with time, spatial localization and other attributes – such approach makes the schema clearer and easier to upgrade.

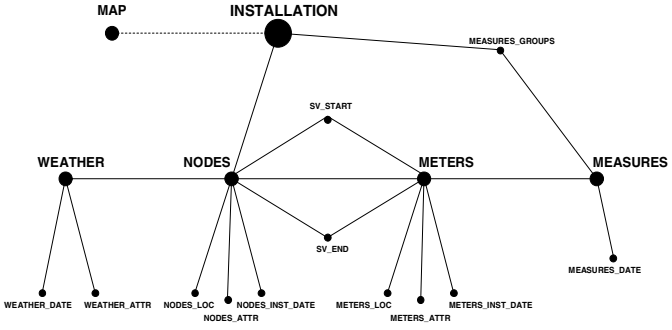


Fig. 3. A schema of a measurement oriented versioned cascaded star for STDW(t)

Table 1. The example of SV_START i SV_END tables

SV_START		*SV_END*	
ID_SV_START	**SV_START_DATE**	**ID_SV_END**	**SV_END_DATE**
1	01/01/2001 00:00:00	1	03/01/2001 15:38:53
2	03/01/2001 15:38:54	2	05/01/2001 10:38:04
3	05/01/2001 10:38:05	3	10/01/2001 13:04:09
4	10/01/2001 13:04:10	4	16/01/2001 01:44:52
5	16/01/2001 01:44:53	5	25/01/2001 19:17:50
6	25/01/2001 19:17:51	6	31/12/2199 00:00:00

To fulfill earlier assumptions, the implemented system should be not only spatial (stores information about objects spatial localization) but also temporal (it has the ability to incorporate modifications of data in time). The temporal character of our model is assured with tables: SV_START and SV_END. These tables store data about structures importance ranges. The tables are used both by NODES and METERS dimensions so it is possible to set importance ranges for nodes and meters. This fact transforms the cascaded star model into fact constellation model in which some tables can be used by several dimensions, just like in the presented model.

Table 1 presents an example of the six structure versions in tables SV_START and SV_END. The chronon between next structures equals one second. The version is created using dates pointed with the same identifiers. For example number 2 structure version is valid through 3 January 2001 15:38:54 to 5 January 2001, 10:38:04. The last structure (no. 6) is valid from 25 January 2001, 19:17:51 up to current moment (NOW). To mark this date we use date 31 December 2199, 00:00:00. The NODES dimension along with its subtables store information about nodes, their attributes, localization and installation dates. Below there is more information about columns:

- NODES_LOC.X, NODES_LOC.Y store information about node localization.
- NODES_ATTR.R stores information about the node area radius. Meters connected with this node are placed in this circular area where the middle point is marked with (NODES_LOC.X, NODES_LOC.Y) and the radius equals NODES_ATTR.R. This results from a fact, that the data transmission is through radio connection, so the node range is appointed by radio range (radius).

Table 2. Values in the END_REASON field in the NODES table

Value	*Description*
D	Node deletion
R	Change of node radius
RT	Change of node type and radius
T	Change of node type
–	No operation – node exists until present moment

Table 3. Example of joint tables – NODES and NODES_ATTR

id_node	*id_sv_start*	*id_sv_end*	*end_reason*	*type*	*r*
2	1	3	D	WG	26
4	1	4	RT	WG	35
4	5	6	–	WGE	40
5	1	6	–	WE	26

- NODES_ATTR.TYPE stores the information about what type of meters are serviced by this node (W, G, E, WG, WE, GE, WGE).
- NODES.END_REASON informs why a certain node was not included in the next version, or what nodes attributes were modified (tab. 2).

From table 3 we can see that, e.g., node no. 2 with the 26 radius and the WG type is still valid in structures 1 – 3 (1 January 2001 00:00:00 to 10 January 2001, 13:04:09, see Tab.1). The validity was lost when the node was deleted (END_REASON = 'D'). Node no. 4 with unchanged attributes (radius=35, type=WG) is still valid in structures 1-4. However, between structures 4/5 the radius is changed (35 → 40) along with the node type (WG → WGE), which is reflected by value 'RT' of END_REASON. After those changes the node is still valid until the present moment (until the end of the structure no. 6 and this is the most actual structure which keeps validity to the point marked as NOW). That is why the value of END_REASON for this node equals '–'. Node no. 5 keeps its value in time for all structures (1-6) until this moment, that is why the value of END_REASON equals '–' (just like in case of node no. 4).

7 Summary

The goal of our work was to design and implement of the multiversion spatio-temporal telemetric data warehouse. We created the working system that uses mechanisms and conception of the aggregate tree and structures versioning, which is based on the cascaded star model. This project can be upgraded and expanded in multiple manners. For example it can be extended into the cascaded star schema with materialized aggregate trees or geographical distribution in Distributed Spatial Telemetric Data Warehouse DSDW(t) [9].

References

1. Baril, X., Bellahsène, Z.: Designing and Managing an XML Warehouse. In: Akmal, B., Chaudhri, A., Zicari, R., Awais Rashid, A. (eds.) XML Data Management. Native XML and XML-Enabled Database Systems, pp. 455–474. Addison-Wesley Professional, Reading (2003)
2. Bebel, B., Eder, J., Koncilia, C., Morzy, T., Wrembel, R.: Creation and Management of Versions in Multiversion Data Warehouse. In: ACM SAC, Nicosia, pp. 717–723 (2004)
3. Bellhase, Z.: Schema Evolution in Data Warehouses. Knowledge and Information Systems 4, 283–304 (2002)
4. Eder, J., Koncilia, C.: Changes of Dimension Data in Temporal Data Warehouses. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) DaWaK 2001. LNCS, vol. 2114, p. 284. Springer, Heidelberg (2001)
5. Golfarelli, M., Lechtenbörger, J., Rizzi, S., Vossen, G.: Schema Versioning in Data Warehouses. In: Wang, S., Tanaka, K., Zhou, S., Ling, T.-W., Guan, J., Yang, D.-q., Grandi, F., Mangina, E.E., Song, I.-Y., Mayr, H.C. (eds.) ER Workshops 2004. LNCS, vol. 3289, pp. 415–428. Springer, Heidelberg (2004)
6. Gorawski, M., Dyga, A.: Indexing of Spatio-Temporal Telemetric Data Based on Adaptive Multi-Dimensional Bucket Index. Fundamenta Informaticae 90(1-2) (2009)
7. Gorawski, M., Gorawski, M.J.: Multiversion spatio-temporal data warehouse. In: Grundspenkis, J., Morzy, T., Vossen, G. (eds.) ADBIS 2009. LNCS, vol. 5739, pp. 291–297. Springer, Heidelberg (2009)
8. Gorawski, M., Malczok, R.: On efficient storing and processing of long aggregate lists. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 190–199. Springer, Heidelberg (2005)
9. Gorawski, M.: Extended Cascaded Star Schema and ECOLAP Operations for Spatial Data Warehouse. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 251–259. Springer, Heidelberg (2009)
10. Jensen, C.S., Dyreson, C.E.: The Consensus Glossary of Temporal Database Concepts. In: Temporal Databases, Dagstuhl (1997)
11. Mendelzon, A.O., Vaisman, A.A.: Temporal Queries in OLAP. In: 26th International Conference on Very Large Data Bases, VLDB 2000, Egypt, pp. 242–253 (2000)
12. Nabil, A., Vijayalakshmi, A., Yesha, Y., Yu, S.: Efficient Storage and Management of Environmental Information. In: IEEE Symposium on Mass Storage Systems (2002)
13. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP Operaions in Spatial Data Warehouses. In: Advances in Spatial and Temporal Databases, 7th International Symposium, SSTD, CA (2001)
14. Papadias, D., Tao, Y., Kalnis, P., Zhang, J.: Indexing Spatio-Temporal Data Warehouses. In: 18th International Conference on Data Engineering, ICDE, San Jose (2002)
15. Tao, Y., Papadias, D.: Historical spatio-temporal aggregation. ACM Transactions on Information, TOIS 23, 61–102 (2005)
16. Vaisman, A.A., Mendelzon, A.O., Ruaro, W., Cymerman, S.G.: Supporting Dimension Updates in an OLAP Server. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T. (eds.) CAiSE 2002. LNCS, vol. 2348, pp. 67–82. Springer, Heidelberg (2002)
17. Wrembel, R., Koncilia, C.: Data Warehouses and OLAP: Concepts, Architectures and Solutions. In: Advances in Data Warehousing and Mining, p. 332. Idea Group, Inc., USA (2007)
18. Wrembel, R., Morzy, T.: Managing and Querying Versions of Multiversion Data Warehouse. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 1121–1124. Springer, Heidelberg (2006)
19. Wrembel, R.: Management of Schema anad Data Evolution in Multiversion data Warehouse. Dissertation 411, Pub. House of Poznan Univ. of Technology, p. 338 (2007)

Indexing Multiversion Data Warehouse: From ROWID-Based Multiversion Join Index to Bitmap-Based Multiversion Join Index

Jan Chmiel

Poznań University of Technology, Institute of Computing Science
and QXL Poland(Allegro.pl)
Jan.Chmiel@allegro.pl

Abstract. This paper presents two index structures, called a Bitmap-based Multiversion Join Index (B-MVJI), designed for the optimization of star queries that access multiple data warehouse versions. The B-MVJI indexes a two-dimensional space data values - data warehouse versions by means of bitmaps. The variant of the B-MVJI, called BS-MVJI, based on sorted bitmaps is also presented. The B-MVJI and BS-MVJI were evaluated experimentally and compared to some alternative approaches.

1 Introduction

A data warehouse (DW) integrates and stores data from external data sources (EDSs). In practice contents and structures of EDSs evolve in time. The evolution of EDSs impacts a DW that has to evolve accordingly. Four main approaches to handling the evolution of DWs have been proposed in the research literature. They can be classified as: (1) schema evolution, e.g., [2,10], (2) temporal extensions, e.g., [17,7,15], (3) versioning extensions [3,9,20], and (4) a Multiversion Data Warehouse Approach, e.g., [27,26] (a comprehensive overview of these approaches can be found in [25]). In the latter approach, the Multiversion Data Warehouse (MVDW) is composed of the sequence of persistent versions, each of which describes a DW schema and data within a given period of time. A *DW version* is in turn composed of a schema version and an instance version.

In a DW, typical types of queries are the so-called *star queries*. They join fact tables with multiple dimension level tables. Reducing execution time of such queries is crucial to a DW performance (for any type of a DW). To this end, a special data structure, called a *join index* was developed [23] that is a B-tree index storing a precomputed join of a fact and dimension level table.

Paper Contribution. In this paper we propose the Bitmap-based Multiversion Join Index (B-MVJI) for indexing data in the MVDW (cf. Section 4). The index is designed for the optimization of star queries accessing multiple DW versions. The B-MVJI is composed of two bitmap indexes that index a two-dimensional space data values-DW versions. The experimentally evaluated efficiency of the B-MVJI shows its promising performance (cf. Section 5).

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 71–78, 2010.

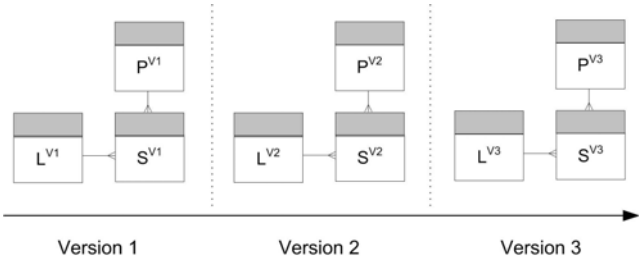


Fig. 1. An example MVDW composed of three versions

2 Motivating Example

In order to present the concept of indexing data in the MVDW, let us consider an example DW that is composed of three DW versions, cf. Figure 1, denoted as $V1$, $V2$, and $V3$. Every DW version is composed of a schema version that, in turn, is composed of respective versions of the *Sales* fact table (denoted as S^{Vi}, i={1, 2, 3}), versions of dimension level table *Location* (denoted as L^{Vi}, i={1, 2, 3}), and versions of dimension level table *Product* (denoted as P^{Vi}, i={1, 2, 3}). A star query accessing data in the three DW versions (further called a multiversion star query) may look as follows:

```
select prodName, shopName, sum(price)
from Sales S, Product P, Location L
where S.prodID=P.prodID and S.locID=L.locID group by prodName, shopName
version in (V1, V2, V3)
```

A traditional (straightforward) technique to support multiversion star queries would be to create separate join indexes (SJI) in each of the three DW versions. Thus, in our example, one should create two indexes (joining *Sales* and *Location* as well as *Sales* and *Product*) in each of the three DW versions. Generalizing our discussion, for n DW versions, n SJI would need to be created.

3 ROWID-Based Multiversion Join Index

In [4] we proposed the ROWID-based Multiversion Join Index (R-MVJI). The index joins multiple versions of a fact table with multiple versions of a dimension level table. Its internal structure combines two indexes, namely a *value index* (ValI) and a *version index* (VerI), cf. Figure 2. Both of them are B^+-tree based. The ValI is created on a join attribute, similarly as a traditional join index. Its leaves store both: (1) values of an indexed attribute (denoted as Key_1, Key_2, ..., Key_n) and (2) pointers to the VerI (denoted as $VIPTR_1$, $VIPTR_2$, ..., $VIPTR_n$). The VerI is used for indexing DWV. Its leaves store lists of ROWIDs, where ROWIDs in one list point to data records (of a fact and a dimension level table) in one DW version. Thus, for a searched value v of a join attribute A, the leaves of *ValI* point to all DWV that store versions of records whose value of attribute A equals v.

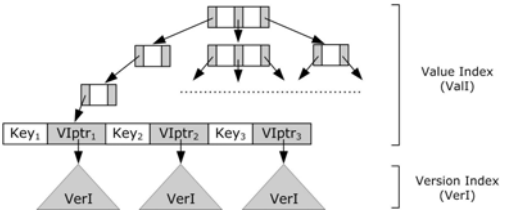


Fig. 2. A schematic view of the structure of the R-MVJI

4 Bitmap-Based Multiversion Join Index

Since the R-MVJI is B^+-tree based, it offers good performance for attributes of wide domains and for queries that access up to maximum 10% of rows. For attributes of narrow domains we developed the Bitmap-based MVJI (B-MVJI). The B-MVJI is composed of two bitmap join indexes. The first one, called a *value bitmap index* (ValBI) is created on a join attribute. The ValBI points to data records in all DW versions that have a given value of an index key. The second index, called a *version bitmap index* (VerBI) is used to index DW versions. Key values Vi (i=1, ..., n) of the VerBI are DW version identifiers. For every index key value, the VerBI points to data records that belong to a given DW version Vi. The structure of the B-MVJI is presented in Figure 3.

A multiversion query can be answered with the support of the B-MVJI as follows. In the first step, the VerBI is accessed in order to compute a bitmap pointing to versions of interest. In the second step, the ValBI is accessed in order to compute a bitmap pointing to records having values of interest. The

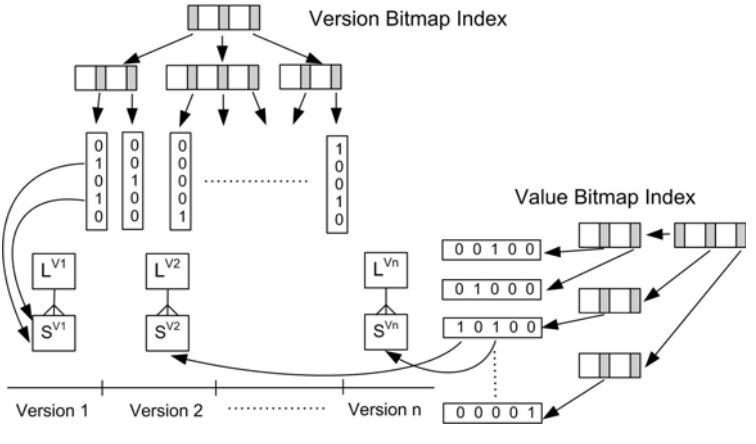


Fig. 3. A schematic view of the structure of the B-MVJI

final bitmap is computed by AND-ing the two bitmaps computed in the first and second step.

Notice that, historical DW versions do not change (except alternative DW versions used for simulation purposes). Therefore, their instances (data) can be ordered by DW version identifiers. Such ordering requires reorganizing data on disk that can be done off-line for historical DW versions. Ordered DW instances can be indexed by the VerBI whose bitmaps are ordered accordingly. Such a variation of the B-MVJI will further be called a Bitmap-based Sorted MVJI (BS-MVJI). The BS-MVJI takes advantage of sorted bitmaps that contain homogeneous vectors of bits equal '1' or '0'. Such bitmaps are more efficient in processing and compressing.

5 Experimental Evaluation

The performance of the B-MVJI, BS-MVJI, R-MVJI, and a traditional approach was evaluated experimentally, for star queries with the pattern shown in Section 2, for two scenarios. In the first one, the number of DW versions accessed by a multiversion query was random. The query selected a constant number of data records from every DW version. In the second scenario, the number of DW versions accessed by a multiversion query was constant, but the selectivity of an indexed attribute was random. The performance measure was the number of index blocks accessed in order to find the answer to a test query. The indexes were implemented in C++. All indexes and indexed data were stored in OS files. The experiments were run on a server machine (8 core Xeon, 16GB RAM) under Linux. The following parameters were set up: the number of DW versions N_{DWV}=100; the number of fact data records in every DW version N_{dr}=100 000; the average size of a single fact data record $Size_{dr}$=64B; the average size of a single record describing a DW version $Size_{vr}$=64B; the size of a pointer to a data record P=32B; the size of an index data key K=32B; the size of a data block B=4096B; data block filling factor $Block_{fill}$=0.75; the number of DW versions accessed N_v: variable from 4 to 100.

5.1 Variable Number of DW Versions

This experiment evaluated the efficiency of the indexes with respect to the number of versions accessed by a multiversion query. The indexed attribute was the primary key of the *Location* table. The selectivity of the attribute $Attr_{sel}$=0.5%. The data records were distributed evenly in the table. Indexes of four different orders p={16, 64, 256, 1024} were tested. The query selected 10% of data records from every DW version. The results are shown in Figure 4.

As we can observe from the charts, the B-MVJI and BS-MVJI perform better (require less block accesses) for lower tree order p, e.g., for p=16, the B-MVJI and BS-MVJI perform much better when the number of DW versions accessed reaches over 30. For p=64, the the BS-MVJI performs better when N_v is lower than 70. For p={256, 1024}, the R-MVJI offers the best performance. Such a

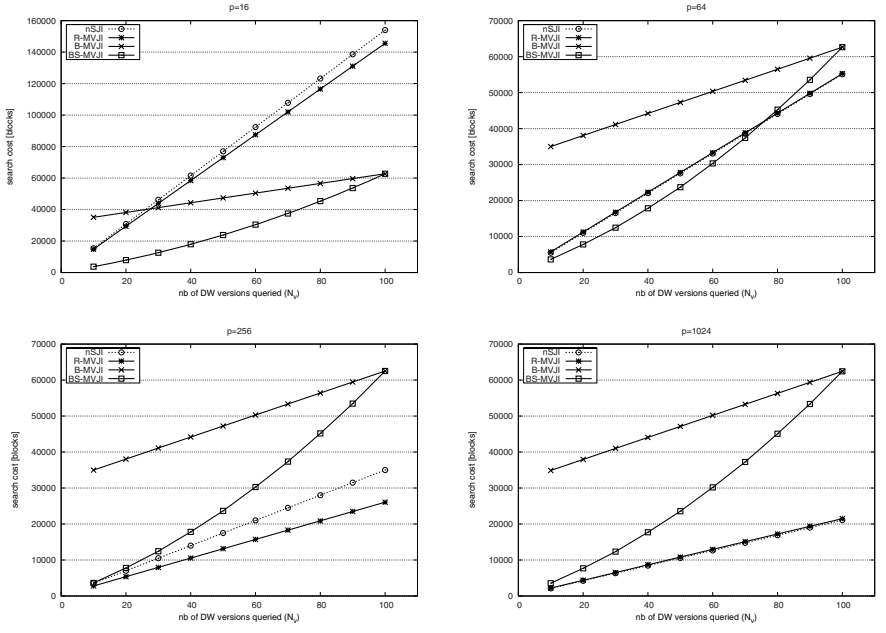


Fig. 4. Variable number of versions accessed by a multiversion query

behavior results from higher values of B$^+$-tree height for smaller p. That, in turn, results in more B$^+$-tree pages that have to be fetched for indexes of smaller p. Bitmap indexes are not impacted by the value of p, therefore their performance characteristic remain constant.

5.2 Variable Selectivity of an Indexed Attribute

This experiment evaluated the efficiency of the indexes with respect to the selectivity of indexed attribute A in table *Sales*, being the foreign key to the *Location* table. The selectivity of the attribute $Attr_{sel}$= {0.5%, 1%, 5%, 10%}. The obtained results are shown in Figure 5.

As we can observe from the charts, the higher value $Attr_{sel}$ of an indexed attribute, the lower number of block reads is required in all of the tested indexing techniques. It is intuitive, since all indexes contain fewer data entries for attributes of higher selectivities. Moreover, the performance of the evaluated indexing techniques depends on tree order. For p=16 the B-MVJI outperforms its competitors within the whole range of tested $Attr_{sel}$. For p={64, 256, 1024} the B-MVJI performs worse than its competitors in the whole range of tested selectivities. The BS-MVJI outperforms its competitors for all the tested values of p. Generally, the B-MVJI and BS-MVJI perform better for higher values of $Attr_{sel}$ since, the higher selectivity value, the less bitmaps need to be stored and processed in the B-MVJI and BS-MVJI.

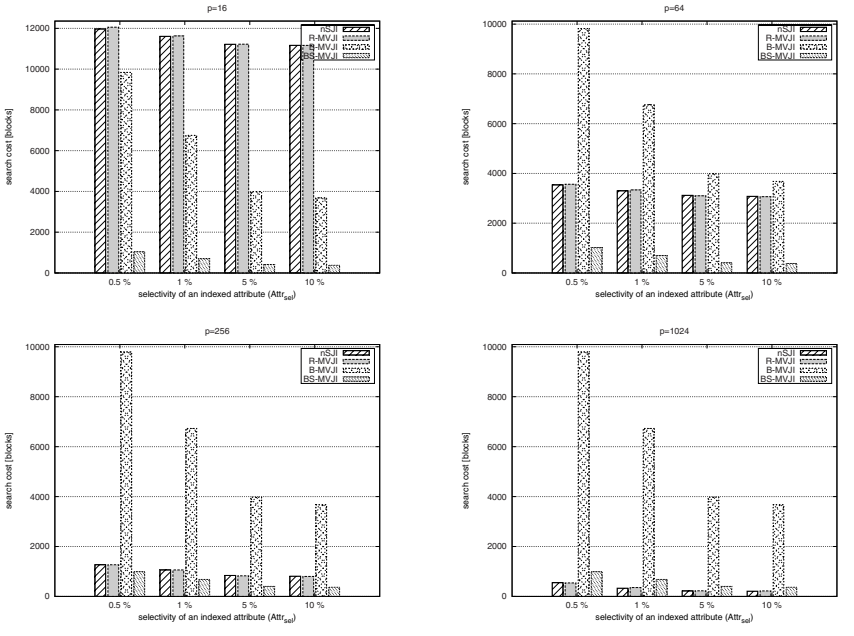


Fig. 5. Variable selectivity of an indexed attribute

6 Related Work

Several indexing techniques for the management of data versions were proposed in the research literature. [8,12,1,24] propose B-tree based indexes for managing temporal versions of data. In [18,19] the authors proposed an indexing technique where time intervals (either valid or transaction) are mapped into a single value which is indexed by a B^+-tree. In [16,22] the authors proposed an index for indexing data records in a 2-dimensional space (transaction time and data value). In [21] the authors proposed an indexing technique for temporal versions of data records whose versions may branch. To this end, a B-tree like structure is used for indexing both data values and database versions. Recently, in [11], three different B^+-tree based index structures for multiversion data were compared analytically and experimentally.

The aforementioned index structures were developed for storing and searching versions of data that are stored in the same table. Moreover, they do not offer means for optimizing queries that join tables. Star query optimization in traditional databases/data warehouses is supported among others by: a join index [23], a bitmap join index [14], a parallel star join [5]. In the latter technique, every foreign key column in the fact table is stored as a partitioned join index whereas all the other columns are stored as a replicated projection index. The indexes and techniques devoted to star query optimization were developed for traditional DWs and cannot be directly applied to the MVDW.

7 Summary

In this paper we proposed the Bitmap-based Multiversion Join Index developed for indexing data in the Multiversion Data Warehouse. The B-MVJI was experimentally evaluated and compared to alternative approaches. The obtained results show that the B-MVJI offers better performance than its competitors for certain ranges of values of tree order, indexed attribute selectivity, and the number of DW versions accessed. We also proposed a variant of the B-MVJI with sorted bitmaps, developed for indexing historical DW versions. Its performance is much better than the B-MVJI. Future work will focus on applying bitmap compression techniques to the B-MVJI and BS-MVJI as well as on query optimization techniques based on the proposed indexes.

References

1. Becker, B., Gschwind, S., Ohler, T., Seeger, B., Widmayer, P.: An asymptotically optimal multiversion B-tree. VLDB Journal 5(4), 264–275 (1996)
2. Blaschka, M., Sapia, C., Hofling, G.: On schema evolution in multidimensional databases. In: Mohania, M., Tjoa, A.M. (eds.) DaWaK 1999. LNCS, vol. 1676, pp. 153–164. Springer, Heidelberg (1999)
3. Body, M., Miquel, M., Bédard, Y., Tchounikine, A.: A multidimensional and multiversion structure for OLAP applications. In: Proc. of ACM Int. Work. on Data Warehousing and OLAP (DOLAP), pp. 1–6 (2002)
4. Chmiel, J., Morzy, T., Wrembel, R.: Multiversion join index for multiversion data warehouse. Information and Software Technology 51, 98–108 (2009)
5. Datta, A., VanderMeer, D., Ramamritham, K.: Parallel star join + dataindexes: Efficient query processing in data warehouses and olap. IEEE Transactions on Knowledge and Data Engineering (TKDE) 14(6), 1299–1316 (2002)
6. Easton, M.: Key-sequence data sets on indelible storage. IBM Journal on Research and Development 30(3), 230–241 (1986)
7. Eder, J., Koncilia, C., Morzy, T.: The COMET metamodel for temporal data warehouses. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T. (eds.) CAiSE 2002. LNCS, vol. 2348, pp. 83–99. Springer, Heidelberg (2002)
8. Elmasri, R., Wuu, G., Kim, Y.J.: Efficient implementation of techniques for the time index. In: Proc. of Int. Conf. on Data Engineering (ICDE), pp. 102–111 (1991)
9. Golfarelli, M., Lechtenbörger, J., Rizzi, S., Vossen, G.: Schema versioning in data warehouses. In: Wang, S., Tanaka, K., Zhou, S., Ling, T.-W., Guan, J., Yang, D.-q., Grandi, F., Mangina, E.E., Song, I.-Y., Mayr, H.C. (eds.) ER Workshops 2004. LNCS, vol. 3289, pp. 415–428. Springer, Heidelberg (2004)
10. Hurtado, C.A., Mendelzon, A.O., Vaisman, A.A.: Updating OLAP dimensions. In: Proc. of ACM Int. Work. on Data Warehousing and OLAP (DOLAP), pp. 60–66 (1999)
11. Jouini, K., Jomier, G.: Indexing multiversion databases. In: Proc. of ACM Conf. on Information and Knowledge Management (CIKM), pp. 915–918 (2007)
12. Lanka, S., Mays, E.: Fully persistent B^+-trees. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 426–435 (1991)
13. Lomet, D., Salzberg, B.: Access methods for multiversion data. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 315–324 (1989)

14. Loney, K.: Oracle Database 11g The Complete Reference. McGraw-Hill/Osborne (2008)
15. Malinowski, E., Zimányi, E.: Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications. Springer, Heidelberg (2008)
16. Manolopoulos, Y., Kapetanakis, G.: Overlapping B^+-trees for temporal data. In: Proc. of Jerusalem Conf. on Inf. Technology (JCIT), pp. 491–498 (1990)
17. Mendelzon, A.O., Vaisman, A.A.: Temporal queries in OLAP. In: Proc. of Int. Conf. on Very Large Data Bases (VLDB), pp. 242–253 (2000)
18. Nascimento, M.A.: A two-stage B^+-tree based approach to index transaction time. In: Proc. of Int. Work. on Issues and Applications of Database Technology (IADT), pp. 513–520 (1998)
19. Nascimento, M.A., Dunham, M.H.: Indexing valid time databases via B^+-trees. IEEE Trans. on Knowledge and Data Engineering (TKDE) 11(6), 929–947 (1999)
20. Rizzi, S., Golfarelli, M.: X-time: Schema versioning and cross-version querying in data warehouses. In: Proc. of Int. Conf. on Data Engineering (ICDE), pp. 1471–1472 (2007)
21. Salzberg, B., Jiang, L., Lomet, D., Barrena, M., Shan, J., Kanoulas, E.: A framework for access methods for versioned data. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 730–747. Springer, Heidelberg (2004)
22. Tzouramanis, T., Manolopoulos, Y., Lorentzos, N.A.: Overlapping B^+-trees: an implementation of a transaction time access method. Data & Knowledge Engineering (DKE) 29(3), 381–404 (1999)
23. Valduriez, P.: Join indices. ACM Trans. on Database Systems (TODS) 12(2), 218–246 (1987)
24. Varman, P., Verma, R.: An efficient multiversion access structure. IEEE Transactions on Knowledge and Data Engineering (TKDE) 3(9), 391–409 (1997)
25. Wrembel, R.: A survey on managing the evolution of data warehouses. International Journal of Data Warehousing & Mining 5(2), 24–56 (2009)
26. Wrembel, R., Bębel, B.: Metadata management in a multiversion data warehouse. In: Spaccapietra, S., Atzeni, P., Fages, F., Hacid, M.-S., Kifer, M., Mylopoulos, J., Pernici, B., Shvaiko, P., Trujillo, J., Zaihrayeu, I. (eds.) Journal on Data Semantics VIII. LNCS, vol. 4380, pp. 118–157. Springer, Heidelberg (2007)
27. Wrembel, R., Morzy, T.: Managing and querying versions of multiversion data warehouse. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 1121–1124. Springer, Heidelberg (2006)

Towards Evolving Constraints in Data Transformation for XML Data Warehousing*

Md. Sumon Shahriar and Jixue Liu

Data and Web Engineering Lab
School of Computer and Information Science
University of South Australia, SA-5095, Australia
shamy022@students.unisa.edu.au, jixue.liu@unisa.edu.au

Abstract. Transformation of data is considered as one of the important tasks in data warehousing and data integration. With the massive use of XML as data representation and exchange format over the web in recent years, transformation of data in XML for integration purposes becomes necessary. In XML data transformation, a source schema and its conforming data is transformed to a target schema. Often, source schema is designed with constraints and the target schema also has constraints for data semantics and consistency. Thus, there is a need to see whether the target constraints are implied from the source constraints in data transformation. Towards this problem, we define two important XML constraints namely XML key and XML functional dependency(XFD). We then use important transformation operations to see if the source constraints are satisfied by the source document, then the target constraints are also satisfied by the target document. Our study is towards the utilization of constraints data integration and data warehousing in XML.

1 Introduction

Transformation of data is an important activity in some data intensive activities such as data integration and data warehousing[1,2]. Specifically in data integration, there is a need to transform a source schema with its conforming data to a target schema. In recent days, with the massive applications of XML[14] over the web, XML data transformation for integration purposes[6,7] becomes important. In XML data transformation[5,3,4], a source XML schema is often designed with XML constraints[11,12,13] to convey semantics and data integrity. Similarly, the XML target schema is also often designed with constraints. Thus after transformation, there is a need to see whether the target constraints are implied from the source constraints as a result of transformation operations. We illustrate the research question in Fig.1. In Fig.1, consider an XML source Document Type Definition(DTD) D_S, its conforming document T_S and valid constraints C_S on D_S. The transformation operation τ has two sub-operations: the schema transformation τ_D and the document transformation τ_T. The operations τ_D produce

* This research supported with Australian Research Council(ARC) Discovery Project(DP) Fund.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 79–86, 2010.

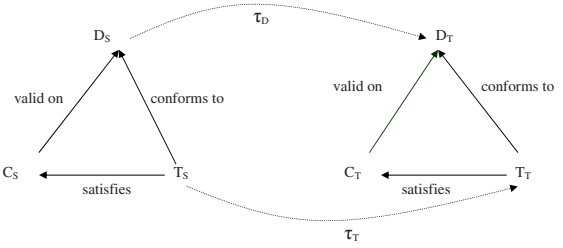


Fig. 1. The Problems

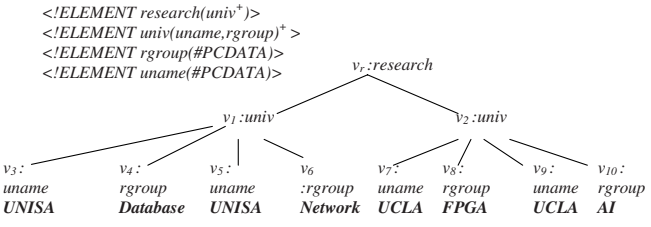


Fig. 2. An XML DTD D_a and the document T_a

the target schema D_T and the operations τ_T produce the target document T_T. Consider the constraints C_T on the target schema D_T. Now our question is: *If T_S satisfies C_S then whether T_T satisfies C_T.*

We now illustrate the research question using a motivating example. Consider D_a as the source DTD and its conforming document T_a in Fig.2. The figure illustrates the research groups of universities. We see that in each *univ* node, there are research group names($rgroup$) with their associated university names($uname$). Now consider the XML functional dependency(XFD) $\Phi_a(research/univ, \{uname\} \rightarrow univ)$ on the DTD D_a meaning that $uname$ determines $univ$. We say XFD Φ_a is satisfied by the document T_a because in each $univ$ node, there is at least one $uname$ element(a technical definition of XFD and its satisfaction[15] will be given later). Note that there is more that one $uname$ under each $univ$ node as functional dependency allows redundant data. If we observe the document T_a, we see that under first $univ$ node v_1, there are two $uname$ nodes v_3, v_5 with the same "$UNISA$" value and under the second $univ$ node v_2, there are two $uname$ nodes v_7, v_9 with the same "$UCLA$" value. If we use $nest(rgroup)$ meaning that research group names with same university are nested, then we get the document T_b. Surely we transform the source DTD D_a to D_b as the target DTD accordingly. Now consider the XML key[16] $\mathbb{k}_b(research/univ, \{uname\})$ on the DTD in Fig.3. Then we see that the document T_b satisfies the key $\mathbb{k}_b$ meaning that for all $univ$ nodes, $uname$ with values "$UNISA$" and "$UCLA$" are distinct.

Observation 1: *XML key is implied from XML functional dependency using nest operation.*

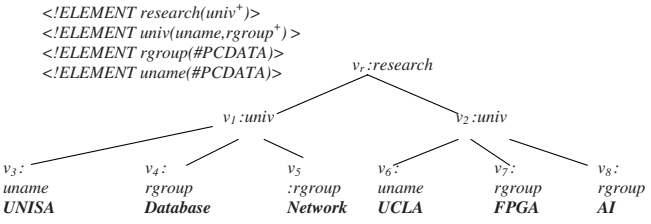


Fig. 3. An XML DTD D_b and the document T_b

While observing the research problems, we aim to achieve the following contributions.

- We define XML key [16] and XFD [15] on DTD and their satisfactions. The definitions for XML key and XFD consider the ordered model of XML data.
- We study how target constraints are implied from source constraints given some transformation operations.
- We finally show the experimental results on satisfactions for XML key and XFD in constraint implications.

2 Basic Definitions and Notation

In this section, we illustrate the definitions for XML key and XFD using examples. For detailed definitions, we refer to [15] for XML key and [16] for XFD. First, we give basic definitions and notation for DTD and the document those are necessary in defining XML key and XFD. A DTD is defined as $D = (EN, \beta, \rho)$ where EN contains element names and ρ is the root of the DTD and β is the function defining the types of elements. For example, the DTD D_a in Fig.2 is defined as $\beta(research) = [univ^+]$, $\beta(univ) = [uname{\times}rgroup^+]$, $\beta(uname) = Str$, $\beta(rgroup) = Str$, $EN = \{research, univ, uname, rgroup, Str\}$, $\rho = research$ and $Str = \#PCDATA$. An element name and a pair of squared brackets '[ ]' each, with its multiplicity, is called a *component*. For example, $[univ^+]$, $[uname{\times}rgroup^+]$ are two components. A sequence of components, often denoted by g, is called a *structure* s.t. $g = [uname{\times}rgroup^+]$. A structure can be further decomposed into *substructures* such as g can be decomposed into $g = g_1{\times}g_2$ and $g_1 = uname$ and $g_2 = rgroup^+$. We note that special cases of structures are components and that multiplicities can only be applied to components as g^c where $c \in [?, 1, +, *]$.

We say $research/univ$ is a complete path and $univ/uname$ is a simple path. The function $beg(research/univ) = research$, $last(univ/uname) = uname$ and $par(uname) = univ$.

Now we define XML key $\mathbb{k}(Q, \{P_1, \cdots, P_l\})$. We say Q as *selector* that is a complete path, $\{P_1, \cdots, P_l\}$ is called $fields$ those are simple paths. All paths P_i are ended with $\#PCDATA$ meaning that $\beta(last(P_i)) = Str$. For example, Consider the key $\mathbb{k}_b(research/univ, \{uname\})$ on D_b in Fig.3. We see $research/univ$ is a complete path, $uname$ is a simple path and $\beta(uname) = Str$.

In defining XML key satisfaction, we need some definitions and notation on XML document. For example, the document T_b in Fig.3 is defined as $T_{v_r} = (v_r : research : T_{v_1}T_{v_2})$, $T_{v_1} = (v_1 : univ : T_{v_3}T_{v_4}T_{v_5})$,$T_{v_2} = (v_2 : univ : T_{v_6}T_{v_7}T_{v_8})$. We then define $T_{v_3} = (v_3 : uname : UNISA)$ and the trees $T_{v_4}, T_{v_5}, T_{v_6}, T_{v_7}, T_{v_8}$ are defined in the same way. We say two trees or a sequence of trees are value equivalent($=_v$) if the node names and their values are the same. For example, $T_x = (x : uname : UNISA)$ and $T_y = (y : uname : UNISA)$ are value equivalent. Now consider $g = [uname{\times}rgroup^+]$. We then say hedge $H^g = T_{v_3}T_{v_4}T_{v_5}$ for node v_1 and $H^g = T_{v_6}T_{v_7}T_{v_8}$ for node v_2. The necessity of hedge is to make the production of values for paths of fields. For example, if the paths of fields are $\{uname, rgroup\}$ in a key, then we need to produce close pair values as $(T_{v_3}T_{v_4})$, $(T_{v_3}T_{v_5})$ for node v_1 and $(T_{v_6}T_{v_7})$, $(T_{v_6}T_{v_8})$ for node v_2. We term these pair-wise values as *tuple*.

In case of key $\Bbbk_b(research/univ, \{uname\})$, we find the tuples (T_{v_3}) for node v_1 and (T_{v_6}) for node v_2 and these tuples are value distinct in the whole document T_b. Thus we say that key $\Bbbk_b$ is satisfied by the document T_b. However the key $\Bbbk_b$ is not satisfied by the document T_a in Fig.2 because there are duplicate tuples, for example T_{v_3} and T_{v_5} for those are the value same.

Now we define XFD $\Phi(S, P \rightarrow Q)$. We say S is the *scope* that is a complete path, P is *determinant*(LHS) that is simple path and Q is *dependent*(RHS) that is also simple path. The path Q can be ϵ(empty) meaning that $P \rightarrow last(S)$. In defining XFD satisfaction, we say that in each scope, if two tuples for paths P are the same, then their corresponding tuples for Q are also the same. For example, in Fig.3, the XFD $\Phi_b(research/univ, \{rgroup\} \rightarrow uname)$ is satisfied by the document T_b, but the XFD $\Phi'_b(research/univ, \{uname\} \rightarrow rgroup)$ is not satisfied by T_b. Consider another XFD $\Phi_a(research/univ, \{uname\} \rightarrow \epsilon)$ on the DTD D_a in Fig.2. The XFD Φ_a is satisfied by the document T_a because there is at least one tuple for path P in each scope *univ*.

3 Implication of XML Keys for Nest Operation

In XML data transformation, different transformation operators are used[5,3,4]. The important transformation operations those are found in most literatures are $Nest$ and $UnNest$. In this section, we study how XML key is implied to the target schema from XFD on the source schema using $Nest$ operation.

Before studying implication, we explain the $Nest$ operation.

Definition 1 (Nest). *The nest operation on g_2 in $[g_1 \times g_2^{c_2}]^c$ is defined as, if $g = [g_1 \times g_2^{c_2}]^c \wedge c \supseteq +$, then $nest(g_2) \rightarrow [g_1 \times g_2^{c_2 \oplus +}]^c$. We say g_1 as comparator and g_2 as collector. The multiplicity operation $c_2 \oplus +$ means the multiplicity whose interval encloses those of c_1 and $+$.*

The nest operator restructures the document and it transforms the the flat structure of the document to the nested structure. The *nest* operator merges the hedges of type construct g_2 (the *collector*) based on the value equivalence of the hedges of type construct g_1 (the *comparator*). For example, given

$\beta(e) = [A_{\times}B_{\times}C_{\times}D]^*$ and $T = (e(A:1)(B:1)(C:2)(D:3)(A:1)(B:2)(C:2)(D:4)(A:1)(B:1)(C:2)(D:4))$, the operator $nest(C_{\times}D)$ combines the hedges of the collector $C_{\times}D$ based on the value equivalence of the hedges of the comparator $A_{\times}B$ and produces $\beta_1(e) = [A_{\times}B_{\times}[C_{\times}D]^+]^*$ and $T_1 = (e(A:1)(B:1)(C:2)(D:3)(C:2)(D:4)(A:1)(B:2)(C:2)(D:4))$. Thus we see that after $Nest$ operation, the values for the collector g_1 in the document becomes distinct.

We get the following theorem for the nest operation.

Theorem 1. *Given the transformation $Nest(g_2)$, an XML key $\mathbb{k}_t(Q, \{P\})$ on the target schema is implied from an XFD $\Phi_s(S, P \rightarrow \epsilon)$ on the source schema if the path P is involved in g_1.*

The proof of the theorem follows the transformation definition of the $Nest$ operation. In XFD Φ_s, the tuples for path P needs to be complete and can have two tuples with same value. If the path P in XFD is involved in the structure g_1, then after transformation using $Nest$, the values for path P become distinct which satisfies the key satisfaction property.

We illustrate the theorem using an example.

Example 1. Consider the XFD $\Phi_a(research/univ, \{uname\} \rightarrow \epsilon)$ on the source DTD D_a in Fig.2. This XFD is satisfied by the document T_a because in the selector node v_1, there are two tuples $(v_3 : uname : UNISA)$ and $(v_5 : uname : UNISA)$ and in the selector node v_2, there are two tuples $(v_7 : uname : UCLA)$ and $(v_9 : uname : UCLA)$. After $Nest(rgroup)$, we see that there is one tuple $(v_3 : uname : UNISA)$ for the node v_1 and there is one tuple $(v_6 : uname : UCLA)$ for node v_2 in Fig.3. Considering $[uname_{\times}rgroup]^+$ where $g_1 = uname$ and $g_2 = rgroup$, the path $uname$ in Φ_a is involved in g_1 and it follows the theorem1. Thus the key $\mathbb{k}_b(research/univ, \{uname\})$ is satisfied by the document T_b in Fig.3.

4 Implication of XFD and XML Key for UnNest Operation

As we mentioned in the previous section that $UnNest$ is one of the important transformation operations, thus we study how XFD and XML key are implied to the target schema from XML keys on the source schema using $UnNest$ operation.

Definition 2 (UnNest). *The unnest operation on g_2 in $[g_1 \times g_2^{c_2}]^c$ is defined as, if $g = [g_1 \times g_2^{c_2}]^c \wedge c_2 = +|*$, then $unnest(g_2) \rightarrow [g_1 \times g_2^{c_2 \ominus +}]^{c \oplus +}$. The multiplicity operation $c_2 \ominus +$ means the multiplicity whose interval equals to the interval of c_2 taking that of $+$ and adding $'1'$.*

The *unnest* operator spreads the hedge of the *comparator* type construct g_1 to the hedges of the *collector* type construct g_2. For example, given $\beta(e) = [A_{\times}B_{\times}[C_{\times}D]^+]^*$ and $T = (e(A:1)(B:1)(C:2)(D:3)(C:2)(D:4)(A:$

$1)(B:2)(C:2)(D:4))$, the operator $unnest(C_{\times}D)$ spreads the hedge of the comparator $A_{\times}B$ to the hedges of the collector $C_{\times}D$ and produces $\beta_1(e) = [A_{\times}B_{\times}[C_{\times}D]]^*$ and $T_1 = (e(A:1)(B:1)(C:2)(D:3)(A:1)(B:1)(C:2)(D:4)(A:1)(B:2)(C:2)(D:4))$. We see that the comparator g_1 is distributed to all g_2. Thus the number of g_2 remains unchanged but the number of g_1 is increased with the same value.

We get the following theorems for $UnNest$ operation.

Theorem 2. *Given the operation $UnNest(g_2)$, an XML key $\Bbbk_t(Q, \{P_1, P_2\})$ on target schema is implied from XML keys $\Bbbk'_s(Q, \{P_1\})$ and $\Bbbk''_s(Q, \{P_2\})$ on the source schema if P_1 of $\Bbbk'_s$ is involved in g_1 and P_2 of $\Bbbk''_s$ is involved in g_2.*

The proof of the theorem follows the definition of the $UnNest$ operation. We illustrate the theorem using an example.

Example 2. Consider D_b as the source DTD, the document T_b as the source document and two keys $\Bbbk'_b(research/univ, \{uname\})$ and $\Bbbk''_b(research/univ, \{rgroup\})$ in Fig.3. Both keys are satisfied by the document T_b. We use $UnNest(rgroup)$ to transform D_b and T_b to D_a as the target DTD and T_a as the target document. Considering $uname_{\times}rgroup^+$ where $g_1 = uname$ and $g_2 = rgroup^+$, we see that path $uname$ of key $\Bbbk'_b$ is involved in g_1 and the path $rgroup$ in key $\Bbbk''_b$ is involved in g_2. This follows the condition of the theorem 2. After $UnNest$, we see that the tuples $(v_3 : uname : UNISA, v_4 : rgroup : database)$ and $(v_5 : uname : UNISA, v_6 : rgroup : Network)$ of node v_1 and the tuples $(v_7 : uname : UCLA, v_8 : rgroup : FPGA)$ and $(v_9 : uname : UCLA, v_{10} : rgroup : AI)$ of node v_2 for paths $uname$ and $rgroup$ are distinct in the document T_a that conforms to D_a. Thus the key $\Bbbk_a(research/univ, \{uname, rgroup\})$ is satisfied by the document T_a in Fig.2.

Theorem 3. *Given the operation $UnNest(g_2)$, an XFD $\Phi_t(S, P \rightarrow \epsilon)$ on the target schema is implied from an XML key $\Bbbk_s(Q, \{P\})$ on the source schema if P is involved in g_1.*

We illustrate the theorem using an example.

Example 3. Consider the D_b as the source DTD, the document T_b as the source document and the XML key $\Bbbk_b(research/univ, \{uname\})$ on D_b. The key $\Bbbk_b$ is satisfied by the document T_b as the tuples for path $uname$ are value distinct in the document. We use $UnNest(rgroup)$ to transform D_b and T_b to D_a as the target DTD and T_a as the target document. Considering $uname_{\times}rgroup^+$ where $g_1 = uname$ and $g_2 = rgroup^+$, we see that path $uname$ of key $\Bbbk_b$ is involved in g_1. This follows the condition of the theorem 3. After $UnNest$, we see that the tuples $(v_3 : uname : UNISA)$ and $(v_5 : uname : UNISA)$ of node v_1 and the tuples $(v_7 : uname : UCLA)$ and $(v_9 : uname : UCLA)$ of node v_2 for paths $uname$ in the document T_a that conforms to D_a. Thus the XFD $\Phi_a(research/univ, \{uname\} \rightarrow \epsilon)$ is satisfied by the document T_a in Fig.2.

5 Performances on Checking Implied XML Keys and XFDs at Target Schema

We have already shown how XML key and XFD can be implied from sources to the target schema when $Nest$ and $UnNest$ transformation operations are used respectively. In this section, we study the performances of checking XML key and XFD satisfactions by the transformed, loaded and integrated data at the target schema. All experiments are implemented in Java using a PC with Intel(R) Centrino Duo CPU T2050 at 1.60GHz, 1.49GB RAM and Microsoft Windows XP.

In Fig.4, we show the key satisfaction time where we fix the number of fields to 4 but varying the number of tuples. We see the significant time is spent for tuple generation while the hashing time is nearly constant. We use Java $Hastable(Key, Value)$ to put the values of tuple to check distinctness incrementally. As the tuple generation time and the hashing time are linear, thus the satisfaction time which is the sum of the tuple generation time and the hashing time is also linear.

In similar way of reasoning, the satisfaction time of checking key in Fig.5 is also linear where we fix the number of tuples to 600K but we vary the number of paths in the key.

We show XFD satisfaction time that is linear in Fig.6 where we fix the number of paths to 3 in the LHS but we vary the number of tuples. In Fig.7, the XFD satisfaction time is also linear where we fix the number of tuples to 500K but we vary the number of paths in LHS for an XFD.

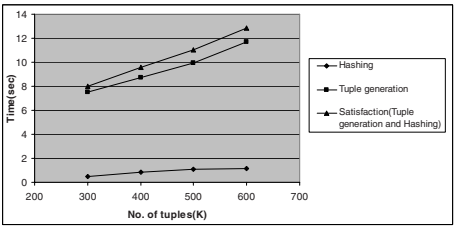


Fig. 4. Key Satisfaction time when the number of fields is fixed to 4

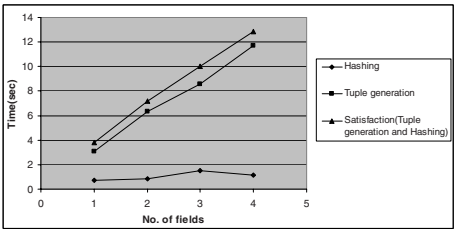


Fig. 5. Key Satisfaction time when the number of tuples is fixed to 600K

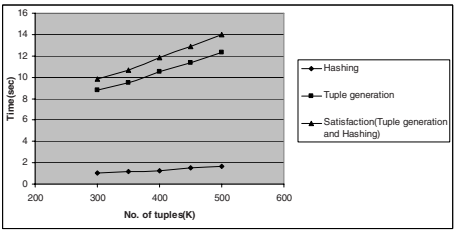


Fig. 6. XFD Satisfaction time when the number of paths in LHS is fixed to 3

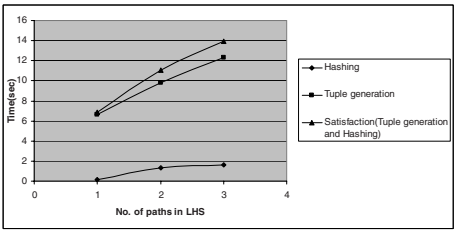


Fig. 7. XFD Satisfaction time when the number of tuples is fixed to 500K

6 Conclusions

We studied the implication of XML constraints in data transformations using important transformation operations namely *Nest* and *UnNest* for constraints implications. In constraints implication, we used our proposed definition for XML key and XML functional dependency and also showed the performances of checking satisfactions of constraints for implication purpose. We further plan to research on how the implications of other XML constraints such as XML inclusion dependency and XML foreign key in XML data integration purposes.

References

1. Fankhouser, P., Klement, T.: XML for Datawarehousing Chances and Challenges. In: Kambayashi, Y., Mohania, M., Wöß, W. (eds.) DaWaK 2003. LNCS, vol. 2737, pp. 1–3. Springer, Heidelberg (2003)
2. Golfarelli, M., Rizzi, S., Vrdoljak, B.: Datawarehouse Design form XML sources. In: DOLAP, pp. 40–47 (2001)
3. Su, H., Kuno, H., Rudensteiner, E.A.: Automating the Transformation of XML Documents. In: WIDM, pp. 68–75 (2001)
4. Erwig, M.: Toward the Automatic Derivation of XML Transformations. In: ER, pp. 342–354 (2003)
5. Liu, J., Park, H., Vincent, M., Liu, C.: A Formalism of XML Restructuring Operations. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 126–132. Springer, Heidelberg (2006)
6. Zamboulis, L., Poulovassilis, A.: Using Automed for XML Data Transformation and Integration. In: DIWeb, pp. 58–69 (2004)
7. Zamboulis, L.: XML Data Integration by Graph Restructuring. In: Williams, H., MacKinnon, L.M. (eds.) BNCOD 2004. LNCS, vol. 3112, pp. 57–71. Springer, Heidelberg (2004)
8. Poggi, A., Abiteboul, S.: XML Data Integration with Identification. In: Bierman, G., Koch, C. (eds.) DBPL 2005. LNCS, vol. 3774, pp. 106–121. Springer, Heidelberg (2005)
9. Li, C.: Describing and utilizing Constraints to Answer Queries in Data Integration Systems. In: IIWeb (2003)
10. Fuxman, A., Miller, R.e.J.: Towards Inconsistency Management in Data Integration Systems. In: IIWeb 2003 (2003)
11. Buneman, P., Fan, W., Simeon, J., Weinstein, S.: Constraints for Semistructured Data and XML. In: SIGMOD Record, pp. 47–54 (2001)
12. Fan, W.: XML Constraints: Specification, Analysis, and Applications. In: DEXA, pp. 805–809 (2005)
13. Fan, W., Simeon, J.: Integrity constraints for XML. In: PODS, pp. 23–34 (2000)
14. Bray, T., Paoli, J., Sperberg-McQueen, C.M.: Extensible Markup Language (XML) 1.0., World Wide Web Consortium (W3C) (February 1998), http://www.w3.org/TR/REC-xml
15. Shahriar, M. S., Liu, J.: Preserving Functional Dependency in XML Data Transformation. In: Atzeni, P., Caplinskas, A., Jaakkola, H. (eds.) ADBIS 2008. LNCS, vol. 5207, pp. 262–278. Springer, Heidelberg (2008)
16. Shahriar, M.S., Liu, J.: Towards the Preservation of Keys in XML Data Transformation for Integration. In: COMAD 2008, pp. 116–126 (2008)

Semantic Optimization of XQuery by Rewriting

Philip Hanson and Murali Mani

Worcester Polytechnic Institute
{phanson,mmani}@wpi.edu

Abstract. Queries on XML data are increasingly widespread in use and scope of application. However, optimization strategies are not yet as developed as they are for traditional DBMSs. Current strategies mostly involve logical or physical query plan optimization. We propose a novel optimization for XQuery using semantic information from the XML schema, where we rewrite a query into an equivalent query with fewer XPath expressions based on schema information. Our experimental results indicate that this optimization can result in substantial performance gains.

1 Introduction

As semistructured data such as XML becomes more prevalent in data storage applications, including RDBMS systems, performance issues specific to semistructured data increase in importance. In this paper, we concentrate on data that has a known structure, specifically XML data with an accompanying DTD or XML Schema definition.

Existing work in schema-aware query optimization was brought to theoretical parity with RDBMS optimization techniques by [12]; [5] discussed optimization through query rewriting. More recent work such as [10] has focused primarily on query simplification through elimination of impossible path expressions and short-circuit evaluation of expressions which always produce the same value. Relatively little has been published regarding other schema-informed optimization.

Our contribution involves the combination of related XPath expressions within an XQuery to simplify the query through rewriting. Like techniques described in [5], our solution modifies the query itself before execution rather than working at the level of logical or physical plan optimization. By combining multiple XPath expressions, we can produce a simpler but equivalent expression that can be evaluated on its own without incurring the overhead of combining multiple overlapping result sets. Because our optimization takes place prior to the creation of a logical/physical plan, it can be used by any XQuery engine and in conjunction with other optimization techniques. For example, our technique is orthogonal to tree-algebra based optimization as in Timber [7], complex query decorrelation [14], sharing of common subpath expressions as in the NEXT framework [3], and various indexing schemes and query evaluation techniques as in [11].

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 87–95, 2010.

Motivating Example: Consider two XQuery expressions:

Q = FOR $r in /building//room, $i in $r//item RETURN $i
Q' = FOR $i in /building//room//item RETURN $i

Depending on the XQuery engine implementation, it could be much more efficient to execute Q' rather than Q. For instance, a naïve implementation for Q might obtain $r and $i separately and do structural joins, whereas for Q' no structural joins are needed. Another use-case in streaming XML scenario is illustrated in our experimental results in Sect. 5.

However, it is not always possible to rewrite Q into Q'. In this paper, we perform reasoning based on the schema to combine XPath expressions so that queries such as Q can be rewritten into Q'. As our experimental results show, the performance benefits of such rewriting can be significant.

Outline: We describe our solution in two steps for easier understanding. How to combine two XPath expressions is described in Sect. 2. The general solution given multiple paths is described in Sect. 3. Section 4 gives the algorithm for our solution, and Sect. 5 describes the experimental observations. Section 6 discusses related work and Sect. 7 concludes the work.

2 Combining Two XPath Expressions

Let us first examine how two given XPath expressions can be combined into a single XPath expression. We use the symbols a, b, c, d, x, and y uniformly in this section to represent definite elements within an XPath expression. Also, we use the expressions P1, P2, P3, and so on to represent subpaths within an XPath expression. These subpaths may be empty.

Problem Definition

Given an XML schema and an XQuery expression of the form

FOR $a in /P1//x/P2/a, $b in $a/P3//P4/b ...

we can rewrite the two variable bindings shown into a single variable binding as *$b in /P1//x/P2/a/P3//P4/b* to improve performance. However, the original variable bindings may include duplicate result nodes, while the revised variable binding will not have any duplicates, due to duplicate elimination inherent to XPath [17]. Note that in this section, we assume that the XPath expression for the second variable binding (*$b*) starts from *$a* and that the *$a* binding is not used in any other path expression in the entire query. Also, we consider only the child axis (/) and the descendant axis (//) in our path expressions [17].

These duplicate results are generated when multiple nodes matched by the first variable binding are ancestors of the same node that is matched by the second variable binding. In order to safely rewrite the query, we must ensure that it is impossible for rewriting to change the result, i.e. that the original query cannot generate duplicate results.

Solution

For ease of writing, we will refer to the first variable binding as the "parent" binding or path and the second as the "child" binding or path. We will call all the nodes in the XML instance that match the child binding the result nodes. Each result node has a path in the instance document from the root to the result node; we call this the "result path". Duplicates occur when the same XML node in the instance document appears multiple times among result nodes. Duplicates occur only when a result path has multiple matches for the parent path (see Examples 1 and 2 below). For the purposes of this paper, we consider the validation of a path against an XML schema to be an algorithmic check against the schema to verify that the path is possible within an XML document that conforms to the schema. This can be accomplished using automaton-based methods as described in [5] or [9]. If a path can be so validated, we say this path is "valid".

Observation 1. *Consider the following parent and child variable bindings:*

Parent: $a = /P1//x/P2/a *; Child:* $b = $a/P3//P4/b

If the path /P1//x/P2/a//x/P2/a/P3//P4/b *can be validated using the XML schema, then it is possible that the original query would return duplicates.*

Note that in Obs. 1 *P1, P2, P3, P4* can be empty and that x can be equal to a.

An informal proof of Obs. 1 is illustrated in Fig. 1. This figure illustrates an instance document as specified by a schema. The condition specified in Obs. 1 is satisfied in this figure; therefore duplicates can be produced. The two a elements bind to *$a* and the b element binds twice to *$b* (one for each of the a bindings), thus producing duplicate result nodes.

Example 1. Consider an XQuery with parent binding *$a = //a* ; child binding *$b = $a//b* . Rewriting it into the form as described in Obs. 1, we get *P1, P2, P3, P4* as empty and $x = a$. To determine whether there will be duplicate result nodes, we need to check whether *//a//a//b* is valid against the schema.

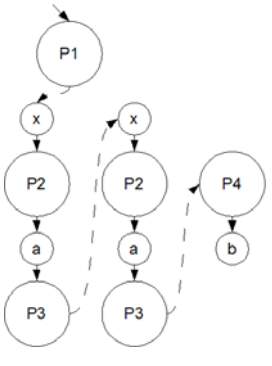


Fig. 1. Instance Document Illustrating Obs. 1

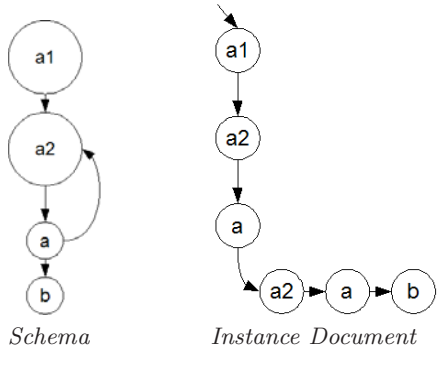


Fig. 2. Document Schema & Instance Document Illustrating Example 2

Note that Obs. 1 specifies, given a parent-child variable binding pair as *$a = P1/a; $b = $a/P2/b* , when there could be duplicate result nodes for the child variable binding. If duplicate result nodes will not be produced, then the two path expressions can be combined into one path expression as: *$b = P1/a/P2/b* and the parent variable binding can be removed; otherwise, no rewriting is possible.

Example 2. Consider an XQuery expression where the parent variable binding is *$a = //a1//a2/a* and the child variable binding is *$b = $a//b* . According to observation 1, we consider the following two options:

Option 1: *P1 =* empty, *x = a1* , *P2 = //a2*
Option 2: *P1 = //a1* , *x = a2* , *P2 =* empty

Let us consider an XML schema such that the path *//a1//a2/a//a2/a//b* is valid, but the path *//a1//a2/a//a1//a2/a//b* is not valid. An instance document is shown in Fig. 2. In this case, while checking Option 1, we determine that the condition specified in Obs. 1 is not satisfied. Of course, we still need to check Option 2, in which case, we determine that duplicates may occur.

The following observation says that we need not check every possible option for P1 and P2 in the parent variable binding; instead, we only need to check one option. In the above example, it is sufficient to check only Option 2. In other words, the following observation states that if Option 1 is valid w.r.to a schema, then Option 2 will necessarily be valid.

Observation 2. *If* P2 *in the expression of Obs. 1 above contains a descendant operator, we can consider* P2 *as* P2 = P5//y/P6 *and reassign* P1, x , *and* P2 *as follows:* P1' = P1//x/P5 *;* x' = y *;* P2' = P6 .
Then P1//x/P2/a//x/P2/a/P3//P4/b *is valid only if* P1'//x'/P2'/a//x'/P2'/a/P3//P4/b *is valid.*

Obs. 2 is obvious, as the rewritten expression is more general than the original form. A similar observation can be made for child variable bindings as below.

Observation 3. *If* P3 *contains a descendant operator,* P3 = P7//P8 *, we can reassign* P3 *and* P4 *as follows:* P3' = P7 *;* P4' = P8//P4 .
Then P1//x/P2/a//x/P2/a/P3//P4/b *is valid only if* P1//x/P2/a//x/P2/a/P3'//P4'/b *is valid.*

We can apply the transformations until *P2* and *P3* have no descendant axis. In Example 2, Option 1 need not be checked; only Option 2 needs to be checked.

3 General Solution

In the previous section, we considered XQuery expressions with a single parent-child variable binding pairing. However, XQuery expressions in general can have several such pairings, which we discuss in this section. Consider the query:

FOR $a in P1, $b in $a/P2, $c in $b/P3, $d in $c/P4, $e in $b/P5 ...

In this query, there are several variable bindings: *$a, $b, $c, $d, $e* ; and several parent-child pairings: ($a, $b), ($b, $c), ($c, $d), ($b, $e). If a path of length zero or more exists from one variable binding to another, we say that there is an ancestor-descendant relationship between them. For the example query, the descendants of *$a* include *$a, $b, $c, $d, $e* . Further, we define ancestor-descendant relationships between pairings and between a pairing and a variable binding. For example, the pairing ($a, $b) is an ancestor of ($c, $d) and also an ancestor of ($a, $b). Similarly, the pairing ($a, $b) is an ancestor of the variable binding $b, and of the variable binding $e.

Observation 4. *Result paths from a given variable binding will become part of result paths of its descendant variable bindings.*

For instance, a result path for $b will be part of a result path of $d.

Observation 5. *If there are duplicates in the result nodes for a binding $z, then there exists an ancestor pairing ($x, $y) that produces duplicates. Note that* $y *and* $z *may be the same.*

Observation 5 says that, if there are duplicates in the result nodes for $c, then duplicates are produced by at least one of the two pairings ($a, $b), ($b, $c).

To identify whether a pairing ($x, $y) produces duplicates, we use Obs. 1, with a small extension. Note that the path expression for $x should now be the "absolute" path starting from the root of the document [17].

Observation 6. *Given a pairing (*$a, $b *) which satisfies the conditions in Obs. 1 (produces duplicates) and a descendant pairing (*$c, $d *) that does not satisfy the condition in Obs. 1 (does not produce duplicates), we can combine $c and $d into one path expression.*

Observation 6 states that, for the example XQuery, if (*$a, $b*) satisfies the condition in Obs. 1, but (*$c, $d*) does not satisfy the condition, then we can rewrite the query as:

FOR $a in P1, $b in $a/P2, $d in $b/P3/P4, $e in $b/P5 ...

4 Algorithm

In light of the above observations and the procedure described in Sect. 3, our rewriting process can be described using the following algorithm:

Inputs: A set of of variable bindings $C_1..C_n$
Outputs: An equivalent set of bindings, with pairs rewritten as a single binding where safe.
Method:

for all pairings (C_i, C_j) **do**
{Let the declarations be C_i = P1; C_j = $$C_i$/P2}
if C_i is not used in any variable binding other than C_j and is not used in the rest of the query **AND** (C_i, C_j) does not produce duplicates by Obs. 1 **then**

Rewrite the path expression for C_j to include path for C_i
{C_j = P1/P2}
Remove the declaration for C_i
end if
{Once rewritten, we have another set of pairings for the whole query and repeat the process until no more pairings can be simplified.}
end for

5 Experimental Results

To evaluate the results of rewriting, we performed a series of experiments based on the XML-to-SQL translation as done in XRel [15]. The XPath conversion mentioned in [15] was extended to process XQuery statements as follows:

Given a query Q containing paths P_1..P_n
For each P_i:
Find all elements matching P_i as set S_i
Perform simple join across all sets S_i to produce result set

Test documents, generated by the XMark XML benchmarking utility [16], were stored in separate but identical database tables, which stored one tuple per element. Each tuple contains the element's name, start index, end index, DOM level, and full path from document root to element. This models the information that is available using a streaming XML engine, as an element's full path can be stored using a stack while processing streaming XML documents.

Using this data model, we can translate an XQuery expression into a SQL statement as mentioned above. For instance, the query *FOR $a in //a, $b in $a//b RETURN $b* can be expressed using the following SQL statement:

```
SELECT n2.name, n2.start, n2.end, n2.level, n2.path
FROM xmark n1, xmark n2
WHERE n1.path LIKE '%.a' AND n2.path LIKE '%.b'
AND n2.start > n1.start AND n2.end < n1.end AND n2.level > n1.level
```

This form can be extended to handle any number of parent-child pairs, including branching queries. Based on Obs. 1, if there are no duplicate nodes in the result for *$b* , then the XML query can be rewritten as *FOR $b in //a//b RETURN $b* , which can be expressed using the following SQL statement.

```
SELECT n2.name, n2.start, n2.end, n2.level, n2.path
FROM xmark n1
WHERE n1.path LIKE '%.a.%.b'
```

SQL statements were executed on a SQL Server 2005 instance running on a test machine with a dual-core AMD Athlon 64 X2 at 2.41 GHz with 2 GB of memory. Fig. 3 shows our first set of experimental results for queries on the 100 MB XMark document. We considered different queries, where each query had different number of parent-child pairings (2 Expressions means the query had 1 parent-child pairing; 3 Expressions means the query had 2 parent-child pairings). Also all these queries produce the same results (irrespective of the number of parent-child pairings). All these queries can be rewritten to have only one path expression. The lower line (Condensed) in Fig. 3, represents the execution time of queries after the

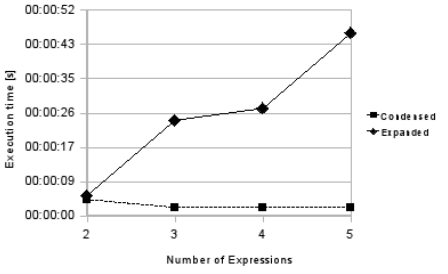


Fig. 3. Execution Time vs. Number of Combined Expressions for 100 MB Document with 2 to 5 Expressions

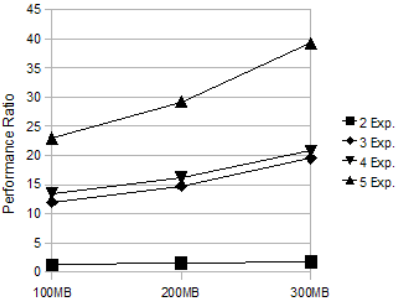


Fig. 4. Performance Ratio, Original vs. Rewritten Queries for 100MB to 300MB Documents with 2 to 5 Expressions

Table 1. Performance Comparison and Ratio of Original and Rewritten Queries for 100MB to 300MB Documents with 2 to 5 Expressions (Relational Database)

	Doc. Size	**2 Exp.**	**3 Exp.**	**4 Exp.**	**5 Exp.**
Original	100 MB	5 s	24 s	27 s	46 s
	200 MB	11 s	74 s	81 s	146 s
	300 MB	16 s	156 s	166 s	314 s
Rewritten	100 MB	4 s	2 s	2 s	2 s
	200 MB	7 s	5 s	5 s	5 s
	300 MB	9 s	8 s	8 s	8 s
Ratio	100 MB	1.25	12	13.5	23
Orig./Rew.	200 MB	1.57	14.8	16.2	29.2
	300 MB	1.78	19.5	20.75	39.25

rewriting as described in Sect. 3. Note that the execution times for the rewritten queries has little variation. This is to be expected, as only a single path comparison is necessary for all the rewritten queries. The second line (Expanded) in Fig. 3 represents the execution times for the queries prior to rewriting, and we can see that the execution time increases with each additional parent-child pairing.

Table 1 gives the actual numbers measured for the original queries and the rewritten queries, for varying number of parent-child pairings, as well as for varying document sizes. It shows that for the same query, the performance benefit of rewriting increases with increasing document size. Fig. 4 illustrates this point. Also, as already shown in Fig. 3, the performance benefit of rewriting increases with increasing number of parent-child pairings that are rewritten.

6 Related Work

Optimization for SQL queries has largely studied logical plan rewriting and physical plan optimization [13], which are mostly outside the scope of this work.

Removing key-foreign key joins can be thought of as query rewriting based on schema constraints, and is done by most commercial database engines such as Oracle. In database theory, rewriting queries by removing unnecessary joins is well studied for conjunctive queries [1]; also utilizing schema constraints such as functional dependencies and inclusion dependencies for removing unnecessary joins is studied.

Semantic rewriting for XQuery at the level of the XQuery core is discussed in [4] with a focus on removal of unused expressions. More involved forms of this method, such as the schema-informed logical query rewriting in [2] check for expressions that always produce the same result or never produce results and prevent their execution through rewriting or elimination. Other techniques such as static type analysis, logical rewriting of core expressions and physical query plan optimizations are also described in [2]. To our knowledge, no work has been published that tries to decrease the number of XPath expressions within a query by combining multiple XPath expressions using semantic constraints. A related work is utilizing schema constraints to determine whether an update specified over an XML view is translatable into updates over the source XML document [8]; here the authors determine whether a source XML element can contribute to multiple view elements.

7 Conclusions and Future Work

Using an algorithm similar to the one described in Sect. 4, it is possible to combine multiple XPath expressions within a query based on the constraints in the given schema. When the schema is not recursive, such rewriting is always possible; when the schema is recursive, rewriting is still possible as long as the schema constraints and the query prohibit duplicates appearing in the result. Our tests indicate that query engines that use an execution model similar to the model described in Sect. 5 will derive a significant benefit in execution time performance using our rewriting. Our approach may also be applicable to streaming query processors in reducing memory footprint. Because our approach is orthogonal to deeper query optimization techniques, it can easily be used in conjunction with other techniques.

Future work might consider further uses of an XML schema to simplify and rearrange specific XPath expressions. In some instances it might be advantageous to split a single expression into multiple disjoint expressions in order to perform optimizations that are possible on only a few of the resultant expressions. Knowledge of the schema is necessary in order to perform such an operation correctly. Another optimization that is worth investigating in the future is to rewrite two XPath expressions into two different XPath expressions while still maintaining the original query semantics, based on the schema constraints. Additionally, some XQuery engines might perform better with more general expressions. Semantic rewriting could be used to generalize expressions that are unnecessarily specific.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Choi, B., Fernández, M., Simeon, J.: The XQuery Formal Semantics: A Foundation for Implementation and Optimization. Bell Labs Research (2002)
3. Deutsch, A., Papakonstantinou, Y., Xu, Y.: The NEXT framework for logical XQuery optimization. In: Proceedings of the Thirtieth international conference on Very large data bases (2004)
4. Fernández, M., Simeon, J., Choi, B., Marian, A., Sur, G.: Implementing XQuery 1.0: The Galax Experience. In: Proceedings of the 29th International Conference on Very Large Data Bases (2003)
5. Fernández, M., Suciu, D.: Optimizing Regular Path Expressions Using Graph Schemas. In: Proceedings of the 14th International Conference on Data Engineering (1998)
6. Fernández, M., Tan, W.C., Suciu, D.: SilkRoute: trading between relations and XML. Computer Networks 33(1-6) (2000)
7. Jagadish, H.V., et al.: Timber: A Native XML Database. The International Journal on Very Large Data Bases 11(4) (2002)
8. Jiang, M., Wang, L., Mani, M., Rundensteiner, E.A.: Updating Views over Recursive XML. In: ICDT Workshop on Emerging Research Opportunities in Web Data Management (2007)
9. Krishnamurthy, R., Chakaravarthy, V., Kaushik, R., Naughton, J.: Recursive XML Schemas, Recursive XML Queries, and Relational Storage: XML-to-SQL Query Translation. In: Proceedings of the 20th International Conference on Data Engineering (2004)
10. Kwong, A., Gertz, M.: Schema-Based Optimization of XPath Expressions. Technical Report, University of California Dept. of Computer Science (2001)
11. Madria, S., Chen, Y., Passi, K., Bhowmick, S.: Efficient processing of XPath queries using indexes. Information Systems 32(1) (2007)
12. McHugh, J., Widom, J.: Query Optimization for XML. In: Proceedings of the International Conference on Very Large Data Bases (1999)
13. Ramakrishnan, R., Gehrke, J.: Database Management Systems, 3rd edn. McGraw-Hill, New York (2003)
14. Shanmugasundaram, J., Kiernan, J., Shekita, E., Fan, C., Funderburk, J.: Querying XML views of relational data. In: Proceedings of the International Conference on Very Large Data Bases (2001)
15. Yoshikawa, M., Amagasa, T., Shimura, T., Uemura, S.: XRel: a path-based approach to storage and retrieval of XML documents using relational databases. ACM Transactions on Internet Technology (2001)
16. XMark XML Benchmarking Project, http://monetdb.cwi.nl/xml/
17. XQuery Home Page, http://www.w3.org/XML/Query/

XCase - A Tool for Conceptual XML Data Modeling*

Jakub Klímek[1], Lukáš Kopenec[1], Pavel Loupal[2], and Jakub Malý[1]

[1] Department of Software Engineering
Faculty of Mathematics and Physics, Charles University in Prague
Czech Republic
jakub.klimek@mff.cuni.cz
[2] Department of Computer Science and Engineering
Faculty of Electrical Engineering, Czech Technical University
Czech Republic
loupalp@fel.cvut.cz

Abstract. Conceptual modeling of XML data was made easier with the introduction of the XSEM model, which utilizes the MDA (Model-driven architecture) ideas of multi-level modeling. XCase is an implementation of this model, enabling users to model their problem domain as a Platform-independent model (PIM), from which Platform-specific models (PSM), XML schemas in our case, can be derived. The main advantage of this approach is maintainability of multiple XML schemas describing the same data from different views as XCase maintains connections between PIM and PSM levels, so that in case of a change to some element, this change can be propagated to all the places where this element is used.

1 Introduction

Recently, eXtensible Markup Language (XML) [16] has become a popular language for data representation. XML assumes data represented in documents whose parts are labeled by marks. Concrete sets of marks (XML formats) are provided by data designers, so they can create various XML formats, each suitable for a particular situation. Exploiting various XML formats in an information system is useful. Each component of the system can process the data in a form that best serves its functions and users. Nonetheless, the existence of a number of XML formats in the system puts several practical questions like how to design the XML formats effectively, how to integrate them in a system and how to maintain them. These questions are fundamental for our work.

Today, data designers use *XML schema languages*, e.g. DTD [16], XML Schema [17] or Relax NG [4], for describing XML formats. An XML format is specified by an

* This work was partially supported by the Ministry of Education of the Czech Republic (grants MSM0021620838 and MSM6840770014) and also by the grant project of the Czech Grant Agency No. GA201/09/0990.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 96–103, 2010.

XML schema that is expressed in one of these languages. However, the languages are unsuitable when designing a set of XML formats that represent the same problem domain. In that case, each XML format is specified by a separate XML schema which leads to modeling certain parts of the domain (e.g. patient) repeatedly for more XML formats (e.g. patient entrance report, surgery record, etc.). This is not only time–consuming and error–prone but also causes problems later when the system evolves. For instance, a new user requirement can result in several changes in more XML schemas. These changes must be done manually by the designer.

In his previous work [11], Nečaský studied how conceptual modeling can help to solve these problems and he proposed a new conceptual model for XML called *XSEM* (Xml SEmantics Modeling). Contribution of this paper is presenting the *XCase* project, an implementation of XSEM.

Motivation. Conceptual modeling for XML has a wide range of application areas that serve as a motivation for our work. *Service–Oriented Architecture* (SOA) [5] is an architectural style for building distributed software systems based on *services* that communicate among each other by messages. The most employed messaging language is XML. SOA can be applied in various domains. Since such domains are heterogeneous, application of XML is profitable. However, interfaces of services in such environments can vary and effective methods for their design, integration and maintenance are important.

Many applications today are in fact web applications based on the client/server pattern. A modern approach today is to apply XML as a communication language between the client and the server. Users usually fill in various types of forms during their work with the application. For each form, a different XML format is applied to send the data to the server. XCase can be directly applied for integration and maintenance in this type of applications as well.

1.1 Roadmap

The rest of this paper is organized as follows. Section 2 surveys the existing approaches to XML schema modeling, describes their main disadvantages and drawbacks and lists possible applications of XCase. Section 3 introduces XSEM model for conceptual XML schema modeling. Section 4 describes XCase including its features and advantages over other existing approaches. Section 5 concludes and suggests future extensions of XCase.

2 Related Work

There exist several approaches to designing XML formats. They can be divided into three categories: *XML schema visualization*, *ER–based approaches*, and *UML–based approaches*. Nevertheless, these approaches all have serious drawbacks described later in this section.

2.1 Model–Driven Architecture

Model-Driven Architecture (MDA) [9] is a general approach to modeling software systems and can be profitably applied to data modeling as well. MDA distinguishes

several types of models that are used for modeling at different levels of abstraction. For our work, two types of models are important. A *Platform-Independent Model* (PIM) allows modeling data at the conceptual level. A PIM diagram is abstracted from a representation of the data in concrete data models such as relational or XML. A *Platform-Specific Model* (PSM) is intended for modeling how the data is represented in a concrete target data model. For each target data model, we need a special PSM that is able to capture its implementation details. A PSM diagram then models a representation of the problem domain in this particular target data model. In other words, it provides a mapping between the conceptual diagram and a target data model schema.

2.2 Approaches to Designing XML

In practice, two conceptual modeling languages are usually considered: *Entity-Relationship Model* (ER) [3] and *Unified Modeling Language* (UML) [12]. The main problem of approaches in the area is that they do not apply MDA sufficiently which brings problems.

ER–based Approaches. ER is for conceptual modeling of relational databases. It contains two modeling constructs. *Entity types* are for modeling real–world concepts. *Relationship types* are for modeling associations among concepts. Both can have attributes that model characteristics of a concept or association. Approaches in this category extend ER to be suitable for conceptual modeling of XML formats. They consider the basic ER constructs and add new ones. *EER* [1] adds constructs for modeling specifics of DTD. *XER* [15] allows modeling specifics of XML Schema. There are also approaches extending ER with constructs that do not strictly follow any target XML schema language. Examples of such approaches are *EReX* [8], *ERX* [13] or *X-Entity* [7].

The authors of these approaches do not consider MDA, but their proposed models are in fact PSMs. This has two negative impacts: (1) At the conceptual level, the designer considers how the data is represented in a XML format instead of considering the data itself. This does not belong to the conceptual level where one should model the domain independently of target XML formats. (2) For two different XML formats two independent conceptual diagrams must be designed without any interrelation. When a concept is represented in both, it must be modeled twice. This makes the conceptual diagrams non–transparent and goes against the principles of conceptual modeling.

UML–based Approaches. UML is a language composed of several sublanguages designed for modeling aspects of software systems. For data modeling, a part called *UML class diagrams* is applied. The basic constructs are *classes* and *associations* whose semantics is similar to ER entity and relationship types. Classes have attributes. Neither ER nor UML can be directly applied for modeling XML formats and must be extended. There already are approaches based on UML [2] [10] [14] which apply MDA. As a PIM they apply the UML class diagrams. As a PSM they propose *profiles*. A profile is a set of *stereotypes* - constructs that can be applied to a construct in a PIM diagram and that specify

how this PIM construct is represented in an XML schema. There are only minor differences in the profiles proposed by the approaches in this group. A typical representative of this approach is Enterprise Architect.

These approaches apply MDA but have significant drawbacks. They are dependent on a certain XML schema language: proposed PSMs are usually intended for XML Schema. Moreover, they consider automatic derivation of PSM diagrams from a PIM diagram. In practice, we need to specify more different XML formats that represent our problem domain for various situations. It would be therefore more practical if a designer could derive more PSM diagrams from the same PIM diagram according to user requirements. This can not be done automatically, manual participation of the designer in the process is necessary.

XML Schema Visualization. This approach is based on visualizing constructs of a particular XML schema language, usually XML Schema, and does not consider MDA at all. It is widely applied in practice and implemented in commercial XML schema design tools, e.g. Altova XML Spy [6]. They do not provide any shift of XML schema languages towards conceptual modeling and do not eliminate problems caused by applying XML schema languages for designing XML formats.

3 Conceptual Modeling with XSEM

XSEM [11] is a conceptual model for XML. It utilizes UML class diagrams to apply MDA to model XML data on two levels: PIM and PSM. For example, a PIM can be a description of a company domain, which usually already exists. A PSM diagram is a visualization of a single XML schema describing a specific type of an XML message used in a company. While the PIM is usually only one, there can be any number of PSM diagrams representing different views on the same company data.

The main feature is that all the XSEM PSM components are formally interrelated with the components of the PIM level. This allows for describing semantics of the PSM components by components from the PIM level. A software implementing XSEM can maintain connections between corresponding PIM and PSM components. These connections enable a change in a PIM component to be propagated to all the affected PSM components in PSM diagrams. Also, a change in a PSM component can be propagated to the PIM level, where all the other derived PSM components can be discovered and updated.

4 XCase

XCase[1] is a tool for conceptual XML data modeling implementing the described XSEM model. Since a tool for conceptual modeling of XML with XSEM has not been developed yet, the main purpose of the project was to examine possibilities of XSEM as well as conceptual modeling for XML in general.

[1] http://www.ksi.mff.cuni.cz/xcase

User work is organized into projects. Each project contains a PIM and a number of PSM diagrams. XCase serves as a full-fledged UML editor. To design PSM diagrams, UML metamodel was extended to support XSEM constructs. Automatic translation of XML formats from their representation as PSM diagrams into XML Schema language is also part of the project.

4.1 Features

Quick and easy to use defining XML formats. XSEM model was designed to visualize XML formats. Working with the visual representation is easier than directly editing the XML Schema files. Moreover, being fully familiar with schema languages is not required.

Avoiding duplications when using the same PIM concepts in different formats. All PSM diagrams in the XCase project are bound to a PIM. Common PIM concepts, their attributes and relations are defined only once in the PIM. When such a concept is created, it can be included in a PSM diagram - a link between the PIM concept and its PSM representation is created.

XML formats design independent of schema language. Modeling XML formats with PSM diagrams is not bound to any specific schema language. Current version of XCase allows users to translate PSM diagrams to XML Schema, but export to other languages such as Relax NG [4] would also be possible.

Consistency checking. During the design process, links between PIM concepts and their representations in PSM diagrams are maintained and can be used to check consistency, to locate usages of PIM concepts in PSM diagrams or to propagate changes. A user can alter both PIM and PSM diagrams at any time without worrying about loss of consistency.

4.2 Platform-Independent Model

PIM enables one to design conceptual diagrams describing the model independently of the intended representation in various XML formats. As a PIM, XCase applies UML class diagrams. Although there is only one PIM in the project, we allow the user to divide it into multiple PIM diagrams to increase readability. In Figure 1(b), there is a PIM diagram of a domain of a company that keeps evidence of purchases. The constructs that are available at the PIM level are the same as defined in the UML class diagrams.

4.3 Platform–Specific Model

A PSM diagram is a visual representation of an XML document structure, so its shape is a forest. XCase supports all the XSEM constructs presented in [11]. A user constructs a desired XML document format from the classes already present in the PIM. This process guarantees that all the PSM components have been *derived* from their conceptual counterparts, maintaining this connection for further use. This includes changes, that can be propagated to all affected components. When a PSM diagram is finished, it can be exported to an XML schema

language. Today, only XML Schema is supported, but there is no problem in exporting to other languages such as Relax NG etc. A more detailed description of the PSM components follows.

PSM Class must be derived from (*represent*) a PIM class. A PSM class models how instances of the represented PIM class are expressed in the XML format. The PSM class has a name and an element label. Root classes of a PSM diagram are created by deriving directly from a PIM class in the PIM. Child PSM classes are added by choosing a PIM path in the PIM from the PIM class represented by the parent PSM class to the desired PIM class to be represented by the new child PSM class. PSM classes are connected by PSM associations.

Structural Representative (SR) is a specific kind of PSM class that *refers* to another PSM class and obtains automatically its attributes and content. The SR extends these obtained components by its own attributes and content. The obtained components are taken into account during the translation to the XML schema. Therefore, SRs allow reusing an already modeled content at more places in a PSM diagram at once. Since PSM diagrams must have a tree structure, we use SRs for modeling recursive structures. Instead of a name of a represented PIM class, the name of a referenced PSM class is displayed.

PSM Attribute belongs to a PSM class. This attribute can either be derived from a represented PIM class attribute, or it can be *PIM-less*, indicating that it only exists in the XML format and not on the conceptual level. Also, a PSM attribute can have an alias - a name that it should have in the XML document.

Attribute Container is used to specify that a set of PSM attributes (now inside the attribute container) is expressed as elements instead of attributes.

Content Container allows for modeling an element that does not have any semantics in terms of the PIM. A PSM content container has a name and has a PSM class as a parent. It models that for each instance of the parent PSM class, the XML code modeled by the components of the container is enclosed in a separate XML element named by the name of the content container.

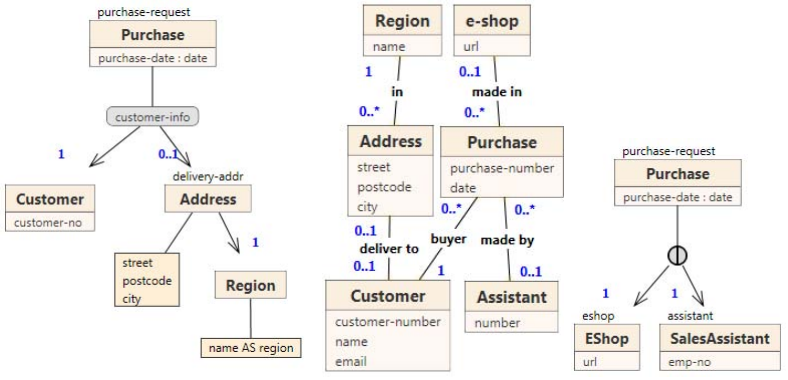
Content Choice models variants in the content of a PSM class. It is assigned to a PSM class and contains PSM associations coming from it. It models that for each instance of the PSM class, only one of the associations is instantiated.

Class Union is an endpoint of a PSM association and contains one or more PSM classes. It models a mixture (i.e. union) of the contained PSM classes. At the instance level, it models a mixture of their instances.

4.4 PSM Examples

In Figure 1(a) there is a PSM diagram of a message representing a purchase, which could be sent to an envelope printer. In Figure 1(c) there is a PSM representation of another message describing the purchase, which could be sent to a counter of purchases made by people and those made via e-shop.

The thing is, that both PSM classes representing the Purchase are connected to the one PIM class Purchase. Therefore, any change to any one of those classes can be propagated to the other two automatically.



(a) PSM diagram of Purchase 1 (b) PIM diagram (c) PSM diagram of Purchase 2

Fig. 1. PIM and PSM diagram examples

4.5 XML Schema Translation

XML formats (represented in XCase as PSM diagrams) describe a set of XML documents. The XSEM representation needs to be translated into one of the schema languages before it can be introduced into a system. XCase currently supports translation from XSEM to XML Schema.

The translation algorithm is automatic and is based on the algorithm proposed in [11], which was fully elaborated to cover all possible PSM diagrams. It uses Venetian Blind[2] design for resulting XML Schema. The XSEM model introduces constructs that provide control over the structure and layout of the conceptual data: *Content container* allows encapsulating some parts of the content under one element. It is translated into an element declaration. *Content choice and class union* are translated to choice content models. *Attributes* in attribute containers are translated to element declarations with simple contents.

5 Conclusions and Future Work

In this paper, we have pointed out current problems with XML data modeling and we analyzed current approaches and among them XSEM, a model for conceptual modeling of XML data. Then we described XCase, a tool implementing XSEM. We described all the constructs used for the two-level modeling in XCase. The algorithm used for XML Schema export is described in detail in XCase documentation.

XCase is currently in use by Fraunhofer Institute for Software and System Engineering (ISST) in Berlin, which has recently given us a positive feedback and also requested some new features. Currently, we work on XML schema evolution and reverse engineering in XCase.

[2] http://www.xfront.com/GlobalVersusLocal.html#ThirdDesign

References

1. Badia, A.: Conceptual Modeling for Semistructured Data. In: Proceedings of the 3rd International Conference on Web Information Systems Engineering Workshops, Singapore, December 2002, pp. 170–177. IEEE Computer Society, Los Alamitos (2002)
2. Bernauer, M., Kappel, G., Kramler, G.: Representing XML Schema in UML - An UML Profile for XML Schema. Technical Report November 2003, Department of Computer Science, National University of Singapore (2003)
3. Chen, P.: The Entity-Relationship Model–Toward a Unified View of Data. ACM Transactions on Database Systems 1(1), 9–36 (1976)
4. Clark, J., Makoto, M.: RELAX NG Specification. Oasis (December 2001), http://www.oasis-open.org/committees/relax-ng/spec-20011203.html
5. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, Upper Saddle River (2005)
6. Altova Inc., XML Spy 2009 (2009), http://www.altova.com
7. Loscio, B., Salgado, A., Galvao, L.: Conceptual Modeling of XML Schemas. In: Proceedings of the Fifth ACM CIKM International Workshop on Web Information and Data Management, New Orleans, USA, November 2003, pp. 102–105 (2003)
8. Mani, M.: Erex: A conceptual model for XML. In: Proceedings of the Second International XML Database Symposium, Toronto, Canada, August 2004, pp. 128–142 (2004)
9. Miller, J., Mukerji, J.: MDA Guide Version 1.0.1. Object Management Group (2003), http://www.omg.org/docs/omg/03-06-01.pdf
10. Narayanan, K., Ramaswamy, S.: Specifications for Mapping UML Models to XML. In: Proceedings of the 4th Workshop in Software Model Engineering, Montego Bay, Jamaica (2005)
11. Nečaský, M.: Conceptual Modeling for XML. Dissertations in Database and Information Systems Series, vol. 99. IOS Press/AKA Verlag (January 2009)
12. Object Management Group. UML Infrastructure Specification 2.1.2 (November 2007), http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF/
13. Psaila, G.: ERX: A Conceptual Model for XML Documents. In: Proceedings of the 2000 ACM Symposium on Applied Computing, Como, Italy, March 2000, pp. 898–903. ACM, New York (2000)
14. Routledge, N., Bird, L., Goodchild, A.: UML and XML Schema. In: Proceedings of 13th Australasian Database Conference (ADC 2002). ACS (2002)
15. Sengupta, A., Mohan, S., Doshi, R.: XER - Extensible Entity Relationship Modeling. In: Proceedings of the XML 2003 Conference, Philadelphia, USA, December 2003, pp. 140–154 (2003)
16. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F.: Extensible Markup Language (XML) 1.0., W3C, 4th edn. (September 2006), http://www.w3.org/TR/REC-xml/
17. Thompson, H.S., Beech, D., Maloney, M., Mendelsohn, N.: XML Schema Part 1: Structures, W3C, 2nd edn. (October 2004), http://www.w3.org/TR/xmlschema-1/

Linear Systems for Regular Hedge Languages

Mircea Marin[1,⋆] and Temur Kutsia[2,⋆⋆]

[1] Department of Computer Science, University of Tsukuba, Japan
[2] RISC, Johannes Kepler University, Linz, Austria

Abstract. We propose linear systems of hedge language equations as a formalism to represent regular hedge languages. This formalism is suitable for several algebraic computations, such as intersections, quotients, left and right factors of regular hedge languages. We indicate algorithms for the translation between hedge automata and linear systems of hedge language equations, and for the computations mentioned before.

1 Introduction

Regular hedge languages (RHLs) play an important rôle in computer science where they are well known as a formalism for a schema of XML [8]. There are many equivalent ways to represent RHLs: by hedge automata [7], regular hedge grammars [6], regular hedge expressions [7], regular expression types for XML [3], etc. The choice of a suitable representation depends on the computation under consideration, and conversions between representations are often required.

We propose a new characterization of RHLs, by linear systems of hedge language equations (LSH for short). LSHs can be viewed as a generalization of the notion of system of linear equations over a Kleene algebra [4] which is linear in both horizontal and vertical directions. An important result is that LSHs have a unique solution and that the solution consists of regular hedge languages. Solving LSHs can be achieved by a slight generalization of solving linear systems over a Kleene algebra. Conversely, for every language L represented by a hedge automaton we can compute an LSH with variables $x_1, \ldots, x_n$ whose solution for x_1 coincides with L. Thus, we can use LSHs to represent RHLs.

LSHs are convenient for several computations in the algebra of RHLs. Many properties of regular word languages carry over to RHLs, such as closure under intersection and quotient, and the fact that the factors of RHLs are regular and finitely many. In this paper we indicate how LSHs can be used to compute the intersection, quotient, left and right factors of regular hedge languages.

The paper is structured as follows. In Sect. 2 we define LSHs and provide algorithms to translate between LSH and hedge automaton. Sections 3–5 describe algorithms for the computation of intersection, right quotient, and left factors of RHLs represented by LSHs. Section 6 concludes.

⋆ Supported by JSPS Grant-in-Aid no. 20500025 for Scientific Research (C).
⋆⋆ Supported by EC FP6 under the project SCIEnce—Symbolic Computation Infrastructure for Europe (Contract No. 026133).

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 104–112, 2010.

2 Linear Systems of Hedge Language Equations

For any set S, by 2^S we mean the set of all subsets of S. For any finite set A we consider the set A^* of all finite words over A, and denote the empty word by ϵ. The set $\mathbf{Reg}(A)$ of regular expressions over A is defined by the grammar $r ::= 0 \mid 1 \mid a \mid r + r \mid r\,r \mid r^\star$ where $a \in A$. We write $[\![r]\!]$ for the usual interpretation of $r \in \mathbf{Reg}(A)$ as a regular language, and $r_1 \doteq r_2$ if $[\![r_1]\!] = [\![r_2]\!]$. The *constant part* $\mathtt{o}(r)$ of r is defined recursively on the structure of r such that it is 1 if $\epsilon \in [\![r]\!]$ and 0 otherwise [1].

Hedges over an alphabet Σ with constants from a set $\mathcal{K}$ are finite sequences of trees produced by the grammar $h ::= \epsilon \mid k \mid a\langle h\rangle\, h$ where $a \in \Sigma$ and $k \in \mathcal{K}$. We denote this set by $\mathcal{H}(\Sigma, \mathcal{K})$. A *hedge language* (HL) is a set of hedges. The *product* of two HLs L and M is the HL $L\,M := \{h\,h' \mid h \in L, h' \in M\}$.

In this paper we consider only HLs with no constants. We also consider an infinite set $\mathcal{X}$ of hedge language variables and regular hedge expressions over Σ and $\mathcal{X}$ generated by $w ::= 0 \mid 1 \mid x \mid a\langle w\rangle \mid w + w \mid w\,w \mid w^\star$ where $a \in \Sigma$ and $x \in \mathcal{X}$. An *assignment* is a mapping σ from variables to HLs. Given an assignment σ, we interpret regular hedge expressions over Σ and $\mathcal{X}$ as follows: $[\![0]\!]_\sigma := \emptyset$, $[\![1]\!]_\sigma := \{\epsilon\}$, $[\![x]\!]_\sigma := \sigma(x)$, $[\![w_1 + w_2]\!]_\sigma := [\![w_1]\!]_\sigma \cup [\![w_2]\!]_\sigma$, $[\![w_1\,w_2]\!]_\sigma := [\![w_1]\!]_\sigma\,[\![w_2]\!]_\sigma$, $[\![a\langle w\rangle]\!]_\sigma := \{a\langle h\rangle \mid h \in [\![w]\!]_\sigma\}$, and $[\![w^\star]\!]_\sigma := \bigcup_{n=0}^{\infty} [\![w]\!]_\sigma^n$ where $[\![w]\!]_\sigma^0 := \{\epsilon\}$ and $[\![w]\!]_\sigma^n := \{h_1 \ldots h_n \mid h_1, \ldots, h_n \in [\![w]\!]_\sigma\}$ for $n \geq 1$. Also, we write $w_1 \doteq_\sigma w_2$ if $[\![w_1]\!]_\sigma = [\![w_2]\!]_\sigma$.

A *hedge automaton* (HA) is a 4-tuple $\mathcal{A} = (\Sigma, \mathcal{Q}, \mathcal{P}, r_1)$ where Σ is the alphabet for hedges, $\mathcal{Q}$ is a finite set of states, $r_1 \in \mathbf{Reg}(\mathcal{Q})$, and $\mathcal{P}$ is a finite set of transition rules of the form $a\langle r\rangle \to \mathtt{q}$ with $\mathtt{q} \in \mathcal{Q}$, $a \in \Sigma$, and $r \in \mathbf{Reg}(\mathcal{Q})$. The language *accepted* by $\mathcal{A}$ is the set $L(\mathcal{A}) := \{h \in \mathcal{H}(\Sigma, \emptyset) \mid h \to_{\mathcal{P}}^* v \wedge v \in [\![r_1]\!]\}$, where $\to_{\mathcal{P}}$ is the transition relation induced by $\mathcal{P}$ on $\mathcal{H}(\Sigma, \mathcal{Q})$. A hedge language is *regular* (RHL) if it is accepted by a hedge automaton.

A *linear system of hedge language equations* (LSH) over a finite alphabet Σ with variables from $\{x_1, \ldots, x_n\}$ is a system of equations of the form

$$x_i = b_i + \ell_{i1}\,x_1 + \ldots + \ell_{in}\,x_n \qquad (1 \leq i \leq n) \tag{1}$$

with ℓ_{ij} sums of elements from $\{a\langle x_l\rangle \mid a \in \Sigma, 1 \leq l \leq n\}$ and $b_i \in \{0, 1\}$ for all $i, j \in \{1, \ldots, n\}$. If $\ell_{ij} \neq 0$ then we say that x_j occurs at *horizontal position* in the right side of the equation of x_i. A *solution* of (1) is an assignment σ for $\mathcal{X} = \{x_1, \ldots, x_n\}$ such that $x_i \doteq_\sigma b_i + \ell_{i1}\,x_1 + \ldots + \ell_{in}\,x_n$ for all $1 \leq i \leq n$.

Solving Linear Systems of Hedge Language Equations. Suppose $\Sigma = \{a_1, \ldots, a_p\}$. We solve (1) in two steps:

Abstraction step. Let $\mathcal{Q} := \{\mathtt{q}_{kl} \mid 1 \leq k \leq p, 1 \leq l \leq n\}$ be a set of fresh symbols. We replace every coefficient $a_i\langle x_j\rangle$ of (1) with $\mathtt{q}_{ij}$. This replacement produces a linear system of equations over the Kleene algebra $\mathbf{Reg}(\mathcal{Q})$:

$$x_i = b_i + m_{i1}\,x_1 + \ldots + m_{in}\,x_n \qquad (1 \leq i \leq n)$$

where m_{ij} are sums of elements from $\mathcal{Q}$, and $b_i \in \{0, 1\}$.

Solving step. Let $\mathcal{P} := \{a_k\langle r_l\rangle \to \mathtt{q}_{kl} \mid 1 \le k \le p, 1 \le l \le n\}$, and compute

$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} := M^\star \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \quad \text{where } M = \begin{pmatrix} m_{11} \ldots m_{1n} \\ \vdots \quad \ddots \quad \vdots \\ m_{n1} \ldots m_{nn} \end{pmatrix}$$

and $M^\star$ is the asterate of matrix M [4].
The unique solution of (11) is $\sigma := \{x_1 \mapsto L_1, \ldots, x_n \mapsto L_n\}$ where, for every $1 \le i \le n$, L_i is the language accepted by the HA $(\Sigma, \mathcal{Q}, \mathcal{P}, r_i)$.

The correctness of this algorithm can be explained as follows. Let μ be the extension of σ to $\mathcal{X} \cup \mathcal{Q}$ with the assignments $\mu(\mathtt{q}_{kl}) := [\![a_k\langle x_l\rangle]\!]_\sigma$ for all $\mathtt{q}_{kl} \in \mathcal{Q}$. Then $x_i \doteq_\mu m_{i1}\, x_1 + \ldots + m_{in}\, x_n + b_i$ for $1 \le i \le n$. Since $[\![m_{ij}]\!]_\mu \subseteq \bigcup_{k=1}^{p} \bigcup_{l=1}^{n} [\![a_k\langle x_j\rangle]\!]_\sigma$ for all i, j, we learn that m_{ij} denote languages of terms, which are ϵ-free HLs. By [5, Lemma 1], we have $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \doteq_\mu M^\star \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$, thus $x_i \doteq_\mu r_i$ for $1 \le i \le n$. This relation shows that the solution of (11) is unique and that, for every $1 \le i \le n$, $[\![x_i]\!]_\sigma$ coincides with the language recognized by the hedge automaton $(\Sigma, \mathcal{Q}, \mathcal{P}, r_i)$ where $\mathcal{P} = \{a_k\langle r_l\rangle \to \mathtt{q}_{kl} \mid 1 \le k \le p, 1 \le l \le n\}$.

Example 1. The equations $x_1 = 0 + (a_1\langle x_1\rangle + a_2\langle x_2\rangle)\, x_1 + a_1\langle x_1\rangle\, x_2$ and $x_2 = 1 + a_2\langle x_2\rangle\, x_2$ form an LSH over signature $\Sigma = \{a_1, a_2\}$ that can be solved as follows. First we abstract the coefficients $a_1\langle x_1\rangle$ and $a_2\langle x_2\rangle$ by replacing them with $\mathtt{q}_{11}$ and $\mathtt{q}_{22}$ respectively. This replacement produces the new system of equations $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = M \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ where $M = \begin{pmatrix} \mathtt{q}_{11} + \mathtt{q}_{22}\ \mathtt{q}_{11} \\ 0 \qquad \mathtt{q}_{22} \end{pmatrix}$. Then

$$M^\star \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} (\mathtt{q}_{11} + \mathtt{q}_{22})^\star\ (\mathtt{q}_{11} + \mathtt{q}_{22})^\star \mathtt{q}_{11} \mathtt{q}_{22}^\star \\ 0 \qquad\qquad\qquad\qquad \mathtt{q}_{22}^\star \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} (\mathtt{q}_{11} + \mathtt{q}_{22})^\star \mathtt{q}_{11} \mathtt{q}_{22}^\star \\ \mathtt{q}_{22}^\star \end{pmatrix}$$

and we conclude that the solution of this LSH is the assignment σ such that $\sigma(x_1) = L((\Sigma, \{\mathtt{q}_{11}, \mathtt{q}_{22}\}, \mathcal{P}, r_1))$ and $\sigma(x_2) = L((\Sigma, \{\mathtt{q}_{11}, \mathtt{q}_{22}\}, \mathcal{P}, r_2))$, where $r_1 := (\mathtt{q}_{11} + \mathtt{q}_{22})^\star \mathtt{q}_{11} \mathtt{q}_{22}^\star$, $r_2 := \mathtt{q}_{22}^\star$, and $\mathcal{P} := \{a_1\langle r_1\rangle \to \mathtt{q}_{11}, a_2\langle r_2\rangle \to \mathtt{q}_{22}\}$. □

Since an LSH has unique solution, we can define the notion of *LSH for a hedge language* L as any LSH whose solution σ assigns language L to the variable that occurs in the left hand side of its first equation.

Converting HA into LSH. Suppose $\mathcal{A} = (\Sigma, \mathcal{Q}, \mathcal{P}, r_1)$ is an HA and $\Sigma = \{a_1, \ldots, a_p\}$. We indicate how to compute an LSH over Σ and a set of variables $\{x_1, \ldots, x_n\}$ such that its unique solution σ has $\sigma(x_1) = L(\mathcal{A})$.

Let $R := \{r_1\} \cup \{r \mid \exists\, a\langle r\rangle \to \mathtt{q} \in \mathcal{P}\}$. It is well known [1] that for any regular expression $r \in \mathbf{Reg}(\mathcal{Q})$ we can compute: (1) a finite set $\partial_{\mathcal{Q}^*}(r)$ of regular expressions in $\mathbf{Reg}(\mathcal{Q}) \setminus \{0\}$, and (2) for every $s \in \partial_{\mathcal{Q}^*}(r)$, a finite set $\mathtt{lf}(s)$ of pairs $\langle \mathtt{q}, s'\rangle \in \mathcal{Q} \times \partial_{\mathcal{Q}^*}(r)$, such that $s \doteq \mathtt{o}(s) + \sum_{\langle \mathtt{q}, s'\rangle \in \mathtt{lf}(s)} \mathtt{q}\, s'$. Thus, if $\{r_1, \ldots, r_n\} := \bigcup_{r \in R} \partial_{\mathcal{Q}^*}(r)$ then $r_i \doteq \mathtt{o}(r_i) + \sum_{\langle \mathtt{q}, r\rangle \in \mathtt{lf}(r_i)} \mathtt{q}\, r$ for $1 \le i \le n$.

Let $\mathcal{X} := \{x_i \mid 1 \leq 1 \leq n\}$ be a set of fresh variables, and the assignment σ of variables from $\mathcal{X} \cup \mathcal{Q}$ such that $\sigma(x_i)$ is the language accepted by $(\Sigma, \mathcal{Q}, \mathcal{P}, r_i)$ for all $1 \leq i \leq n$, and $\sigma(\mathtt{q})$ is the language accepted by $(\Sigma, \mathcal{Q}, \mathcal{P}, \mathtt{q})$, for all $\mathtt{q} \in \mathcal{Q}$. If we replace every horizontal occurrence of r_i with x_i in the previous equations, we obtain $x_i \doteq_\sigma b_i + \sum_{j=1}^{n} m_{ij}\, x_j$ for $1 \leq i \leq n$, where $b_i = \mathtt{o}(r_i) \in \{0, 1\}$ and m_{ij} are sums of elements of $\mathcal{Q}$ for all $1 \leq i, j \leq n$. We define the regular hedge expressions $\mathtt{re}(\mathtt{q}) := \sum_{(a\langle r_i\rangle \to \mathtt{q}) \in \mathcal{P}} a\langle x_i\rangle$ for all $\mathtt{q} \in \mathcal{Q}$, and $\mathtt{re}(m_{ij}) := \sum_{\mathtt{q} \in S_{ij}} \mathtt{re}(\mathtt{q})$ where S_{ij} is the subset of $\mathcal{Q}$ for which $m_{ij} = \sum_{\mathtt{q} \in S_{ij}} \mathtt{q}$. Then obviously $m_{ij} \doteq_\sigma \mathtt{re}(m_{ij})$ for all $i, j \in \{1, \ldots, n\}$, and thus we have $x_i \doteq_\sigma b_i + \sum_{j=1}^{n} \mathtt{re}(m_{ij})\, x_j$ for $1 \leq i \leq n$. Since $\mathtt{re}(m_{ij})$ are sums of regular hedge expressions from $\{a\langle x\rangle \mid a \in \Sigma, x \in \mathcal{X}\}$, what we've got is an LSH over Σ with variables $x_1, \ldots, x_n$ whose unique solution is the restriction of σ to $\mathcal{X}$. The first component of the solution of this LSH is $\sigma(x_1)$, which is $L(\mathcal{A})$.

Example 2. Consider the HA $\mathcal{A} = (\Sigma, \{\mathtt{q}_{11}, q_{22}\}, \mathcal{P}, (\mathtt{q}_{11} + \mathtt{q}_{22})^\star \mathtt{q}_{11} \mathtt{q}_{22}^\star)$ where $\Sigma = \{a_1, a_2\}$ and $\mathcal{P} = \{a_1\langle(\mathtt{q}_{11} + \mathtt{q}_{22})^\star \mathtt{q}_{11} \mathtt{q}_{22}^\star\rangle \to \mathtt{q}_{11}, a_2\langle \mathtt{q}_{22}^\star\rangle \to \mathtt{q}_{22}\}$.

This is the HA computed in Example 1 from an LSH with 2 equations. In this example we have $R = \{r_1, r_2\}$ where $r_1 := (\mathtt{q}_{11} + \mathtt{q}_{22})^\star \mathtt{q}_{11} \mathtt{q}_{22}^\star$ and $r_2 := \mathtt{q}_{22}^\star$, and $\partial_{\mathcal{Q}^*}(r_1) \cup \partial_{\mathcal{Q}^*}(r_2) = R$. We have $\mathtt{o}(r_1) = 0$, $\mathtt{o}(r_2) = 1$, $r_1 \doteq (\mathtt{q}_{11} + \mathtt{q}_{22})\, r_1 + \mathtt{q}_{11}\, r_2$, $r_2 \doteq \mathtt{q}_{22}\, r_2 + 1$, and $\mathtt{re}(\mathtt{q}_{11} + \mathtt{q}_{22}) = \mathtt{re}(\mathtt{q}_{11}) + \mathtt{re}(\mathtt{q}_{22}) = a_1\langle x_1\rangle + a_2\langle x_2\rangle$, $\mathtt{re}(\mathtt{q}_{11}) = a_1\langle x_1\rangle$, $\mathtt{re}(\mathtt{q}_{22}) = a_2\langle x_2\rangle$. We obtain the equations $x_1 = 0 + (a_1\langle x_1\rangle + a_2\langle x_2\rangle)\, x_1 + a_1\langle x_1\rangle\, x_2$ and $x_2 = 1 + a_2\langle x_2\rangle\, x_2$ which form an LSH whose unique solution σ satisfies the condition that $\sigma(x_1)$ is the language of $\mathcal{A}$. □

3 Intersection of Regular Hedge Languages

In this section we indicate how to compute an LSH for $L \cap M$ from LSHs for L and M. Let's assume given an LSH S made of equations $x_i = c_i + \sum_{k=1}^{m} a_{ik}\, x_k$ $(1 \leq i \leq m)$ and with solution σ such that $\sigma(x_1) = L$, and an LSH T made of equations $y_j = d_j + \sum_{l=1}^{n} b_{jl}\, y_l$ $(1 \leq j \leq n)$ with solution τ such that $\tau(y_1) = M$, and that $c_i, d_j \in \{0, 1\}$, a_{ik} are sums of elements from $\{a\langle x_u\rangle \mid a \in \Sigma, 1 \leq u \leq m\}$, and b_{jl} are sums of elements from $\{a\langle y_v\rangle \mid a \in \Sigma, 1 \leq v \leq n\}$. The idea of computing an LSH for $L \cap M$ is based on the principle of intersecting equations of S with equations of T. When we intersect $x_i = c_i + \sum_{k=1}^{m} a_{ik}\, x_k$ with $y_j = d_j + \sum_{l=1}^{n} b_{jl}\, y_l$, we aim at computing an equation that characterizes the intersection of RHLs $\sigma(x_i) \cap \mu(y_j)$. We regard the set of expressions $\mathcal{Z} := \{x_k \cap y_l \mid 1 \leq k \leq m, 1 \leq l \leq n\}$ as variables and consider the assignment ν for variables from $\mathcal{Z}$ defined by $\nu(x_k \cap y_l) := \sigma(x_k) \cap \mu(y_l)$ for all $1 \leq k \leq m$ and $1 \leq l \leq n$. Since $x_i \doteq_\sigma c_i + \sum_{k=1}^{m} a_{ik}\, x_k$ and $y_j \doteq_\mu d_j + \sum_{l=1}^{n} b_{jl}\, y_l$, we can compute regular hedge expressions s_{ijkl} such that $x_i \cap y_j \doteq_\nu \min(c_i, d_j) + \sum_{k=1}^{m} \sum_{l=1}^{n} s_{ijkl}\, (x_k \cap y_l)$ for $1 \leq i \leq m$ and $1 \leq j \leq n$, where s_{ijkl} are sums of regular hedge expressions of the form $a\langle z\rangle$ with $a \in \Sigma$ and $z \in \mathcal{Z}$. More precisely:

- We identify two families of finite sets $\{U_{ik} \mid 1 \leq i, k \leq m\} \in 2^{\Sigma \times \{x_1, \ldots, x_m\}}$ and $\{V_{jl} \mid 1 \leq j, l \leq n\} \in 2^{\Sigma \times \{y_1, \ldots, y_n\}}$ such that $a_{ik} = \sum_{\langle a, u\rangle \in U_{ik}} a\langle x_u\rangle$ and $b_{jl} = \sum_{\langle a, v\rangle \in V_{jl}} a\langle y_v\rangle$ for all $1 \leq i, k \leq m$ and $1 \leq j, l \leq n$.
- We define $s_{ijkl} := \sum_{a \in \Sigma} \sum_{\langle a, x_u\rangle \in U_{ik} \wedge \langle a, y_v\rangle \in V_{jl}} a\langle x_u \cap y_v\rangle$.

For example, the intersection of the equations $x_1 = 1 + (a\langle x_1\rangle + b\langle x_3\rangle)\, x_1 + (b\langle x_3\rangle + d\langle x_4\rangle)\, x_2$ and $y_2 = 1 + (a\langle y_1\rangle + c\langle y_2\rangle)\, y_1 + b\langle y_4\rangle\, y_2$ produces the equation $x_1 \cap y_2 = 1 + a\langle x_1 \cap y_1\rangle\, x_1 \cap y_1 + b\langle x_3 \cap y_4\rangle\, x_1 \cap y_2 + b\langle x_3 \cap y_4\rangle\, x_2 \cap y_2$.

We can construct an LSH I for the HL $L \cap M = [\![x_1 \cap y_1]\!]_\nu$ as follows:

1. Intersect the first equation of S with the first equation of T and add it to I. This intersection produces an equation with variable $x_1 \cap y_1$ to the left.
2. For every variable $x_k \cap y_l$ that occurs in the right side of some equation already in I, add to I the intersection of the equation for x_k in S with the equation for y_l in T.

This process will terminate because $\mathcal{Z}$ is a finite set, so we can not add indefinitely equations to I. We end up with an LSH of at most $m \times n$ equations for the RHL $[\![x_1 \cap y_1]\!]_\nu = L \cap M$.

4 Quotient of Regular Hedge Languages

The *quotient* of an HL L with respect to an HL M is the HL $M^{-1}L := \{h \mid \exists h' \in M$ such that $h'\, h \in L\}$. Like for regular languages, we can prove that if L is RHL and M is *any* HL then $M^{-1}L$ is RHL. To see why this is so, assume L is the language recognized by an HA $(\Sigma, \mathcal{Q}, \mathcal{P}, r)$ and let $\{r_1, \ldots, r_n\} := \bigcup_{w \in \mathcal{Q}^*} \partial_w(r)$. It can be shown for any hedge h, the HL $\{h\}^{-1}L$ is recognized by an HA from $\left\{(\Sigma, \mathcal{Q}, \mathcal{P}, \sum_{s \in Q'} s) \;\middle|\; Q' \subseteq \{r_1, \ldots, r_n\}\right\}$. This is a finite set of at most $2^{\|r\|+1}$ HAs, where $\|r\|$ is the alphabetic width of $r \in \mathbf{Reg}(\mathcal{Q})$ [1, Corollary 10]. Thus, $\{\{h\}^{-1}L \mid h \in M\}$ is a finite set of RHLs. But $M^{-1}L = \bigcup_{h \in M} \{h\}^{-1}L$ is a finite union of RHLs, hence it is RHL too.

Similarly, we can define the *right quotient* of an HL L with respect to an HL M as the HL $LM^{-1} := \{h \mid \exists h' \in M$ such that $h\, h' \in L\}$. If we define the *symmetric* $L^{\mathtt{s}}$ of L as the language obtained by reversing the order of trees at the outermost level in hedges, then $(L^{\mathtt{s}})^{\mathtt{s}} = L$ and $M^{-1}L = (L^{\mathtt{s}}(M^{\mathtt{s}})^{-1})^{\mathtt{s}}$ for any HLs L and M. Moreover, if L is RHL then $L^{\mathtt{s}}$ is RHL too. Since $M^{-1}L = (L^{\mathtt{s}}(M^{\mathtt{s}})^{-1})^{\mathtt{s}}$, we can achieve quotient computations via right quotient computations. Therefore, in the remainder of this section we consider only the computation of right quotient.

If M is RHL then we can compute a representation of LM^{-1}. In the remainder of this section we indicate a method to compute an LSH for LM^{-1} when we know an LSH for L and an LSH for M.

Suppose the LSHs for L and M are S and T like in the previous section, $\mathcal{X} := \{x_1, \ldots, x_m\}$, $\mathcal{Y} := \{y_1, \ldots, y_n\}$, and let σ and μ be their unique solutions. We can construct an LSH S for LM^{-1} as follows:

1. Let $\to^*$ be the be the reflexive-transitive closure of relation $\to$ defined by
   $x_i \to x_j$ if x_j occurs at horizontal position in the right side of the i-th equation of S,
   and $\{i_1, \ldots, i_p\} := \{i \mid x_1 \to^* x_i\}$. Since $x_1 \to^* x_1$, we can assume $p \geq 1$ and $i_1 = 1$. Let $\mathcal{Z} := \{z_{i_1}, \ldots, z_{i_p}\}$ be a set of p fresh variables and ν be

the assignment for $\mathcal{X} \cup \mathcal{Z}$ which extends σ by associating every z_{i_j} with the hedge language $[\![x_{i_j}]\!]_\sigma [\![y_1]\!]_\mu^{-1}$ for all $j \in \{1, \ldots, p\}$.

2. Since $x_{i_j} \doteq_\sigma c_{i_j} + \sum_{k=1}^{p} a_{i_j i_k}\, x_{i_k}$ for all $j \in \{1, \ldots, p\}$, we can multiply it to the right with $[\![y_{i_j}]\!]_\mu^{-1}$ and obtain $z_{i_j} \doteq_\nu e_{i_j} + \sum_{k=1}^{p} a_{i_j i_k}\, z_{i_k}$ where $e_{i_j} = 1$ if $\epsilon \in [\![x_{i_j}]\!]_\sigma [\![y_1]\!]_\mu^{-1}$ and $e_1 = 0$ otherwise. In this way we obtain the equations

$$z_{i_j} = e_{i_j} + \sum_{k=1}^{p} a_{i_j i_k}\, z_k \qquad (1 \le j \le p)$$

which together with the equations of S constitute an LSH with solution ν.

To compute $e_{i_1}, \ldots, e_{i_p}$ we note that for every $1 \le j \le p$ we have $e_{i_j} = 1$ iff $\epsilon \in [\![x_1]\!]_\sigma [\![y_{i_j}]\!]_\mu^{-1}$ iff $[\![x_1]\!]_\sigma \cap [\![y_{i_j}]\!]_\mu \neq \emptyset$. Thus, it is sufficient to be able to decide for every $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$ whether $[\![x_i]\!]_\sigma \cap [\![y_j]\!]_\mu \neq \emptyset$. Since $[\![x_i]\!]_\sigma = [\![c_i]\!] \cup \bigcup_{k=1}^{m} [\![a_{ik}]\!]_\sigma [\![x_k]\!]_\sigma$ and $[\![y_j]\!]_\mu = [\![d_j]\!] \cup \bigcup_{l=1}^{n} [\![b_{jl}]\!]_\mu [\![y_l]\!]_\mu$, we learn that $[\![x_i]\!]_\sigma \cap [\![y_j]\!]_\mu \neq \emptyset$ iff

1. $c_i = d_j = 1$ (in this case, $\epsilon \in [\![x_i]\!]_\sigma \cap [\![y_j]\!]_\mu$), or
2. there exist $k \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, n\}$ such that $[\![a_{ik}]\!]_\sigma \cap [\![b_{jl}]\!]_\mu \neq \emptyset$ and $[\![x_k]\!]_\sigma \cap [\![y_l]\!]_\mu \neq \emptyset$.

It follows that $[\![x_u]\!]_\sigma \cap [\![y_v]\!]_\mu \neq \emptyset$ iff the judgment $x_u \diamond y_v$ can be inferred with

$$\frac{[c_i = 1 \wedge d_j = 1]}{x_i \diamond y_j} \qquad \frac{x_{i_1} \diamond y_{j_1} \quad x_{i_2} \diamond y_{j_2} \quad [a\langle x_{i_1}\rangle\, x_{i_2} \in \mathtt{rs}(x_i) \wedge a\langle y_{j_1}\rangle y_{j_2} \in \mathtt{rs}(y_j)]}{x_i \diamond y_j}$$

where $a\langle x_{i_1}\rangle\, x_{i_2} \in \mathtt{rs}(x_i)$ means that $a\langle x_{i_1}\rangle\, x_{i_2}$ occurs in the right side of the equation for x_i, and the meaning of $a\langle y_{j_1}\rangle\, y_{j_2} \in \mathtt{rs}(y_i)$ is that $a\langle y_{j_1}\rangle\, y_{j_2}$ occurs in the right hand side of the equation for y_j. In particular $e_i = 1$ iff the judgment $x_i \diamond y_1$ can be inferred with the inference rules mentioned above.

Example 3. Consider the LSHs

$$\begin{aligned} x_1 &= 1 + a_1\langle x_2\rangle\, x_1 + a_2\langle x_4\rangle\, x_2 \\ x_2 &= 1 + a_2\langle x_4\rangle\, x_2 \\ x_3 &= 0 + (a_2\langle x_4\rangle + a_3\langle x_4\rangle)\, x_4 \\ x_4 &= 1 + a_1\langle x_4\rangle\, x_4 \end{aligned} \qquad \begin{aligned} y_1 &= 1 + (a_1\langle y_1\rangle + a_2\langle y_2\rangle + a_3\langle y_1\rangle)\, y_1 \\ y_2 &= 0 + (a_1\langle y_1\rangle + a_2\langle y_2\rangle + a_3\langle y_1\rangle)\, y_3 \\ y_3 &= 1 \end{aligned}$$

with solutions σ and μ. Our construction of an LSH for $[\![x_1]\!]_\sigma [\![y_1]\!]_\mu^{-1}$ yields the LSH with the equations $z_1 = e_1 + a_1\langle x_2\rangle\, z_1 + a_2\langle x_4\rangle\, z_2$ and $z_2 = e_2 + a_2\langle x_4\rangle\, z_2$ besides the equations of the first LSH, where e_1 and e_2 are still to be computed. In this example, $e_1 = e_2 = 1$ because the inference rules

$$\frac{}{x_1 \diamond y_1} \qquad \frac{x_4 \diamond y_2 \quad x_2 \diamond y_3}{x_1 \diamond y_2} \qquad \frac{x_4 \diamond y_1 \quad x_4 \diamond y_3}{x_4 \diamond y_2} \qquad \frac{}{x_2 \diamond y_3} \qquad \frac{}{x_4 \diamond y_1} \qquad \frac{}{x_4 \diamond y_3}$$

are available to infer the judgments $x_1 \diamond y_1$ and $x_1 \diamond y_2$. □

5 Left and Right Factors of a Regular Hedge Language

The following are straightforward generalizations to HLs of notions from Conway's theory of factorizations of regular languages [2]. A product of HLs $F_1 \dots F_n$ is a *subfactorization* of a an HL E if and only if $F_1 \dots F_n \subseteq E$. The languages $F_1, \dots, F_n$ are called the *terms* of the subfactorization. A term F_i is *maximal* if it can not be increased without violating the HL inclusion. A *factorization* of E is a subfactorization in which every term is maximal. A subfactorization $F'_1 \dots F'_n$ of E *dominates* another subfactorization $F_1 \dots F_n$ of E if $F_i \subseteq F'_i$ for all $1 \leq i \leq n$. A *factor* of E is any term of some factorization of E. A *left* (resp. *right*) *factor* of E is one which can be the leftmost (resp. rightmost) term in some factorization of E.

RHLs have finitely many factors. E.g., we can reason as follows to show that the right factors of an RHL E are finitely many: F is right factor of E iff there is a factorization $G\,F$ of E iff $F = \bigcap_{h \in G} \{h\}^{-1} E$ for some hedge language G. We noticed that $\{\{h\}^{-1}E \mid h \text{ a hedge}\}$ is a finite set of RHLs. Therefore, the right factors of E are intersections of RHLs taken from a finite set. Hence, they are RHLs (because RHLs are closed under intersection) and finitely many.

Note that F is a left factor of an HL E if and only if $F^{\mathtt{s}}$ is a right factor of the symmetric language $E^{\mathtt{s}}$. This property enables to conclude that the set of right factors of an RHL is finite too, and to reduce the computation of left factors of an RHL to a computation of right factors of an RHL and vice versa.

From now on we consider only the problem of computing LSHs for the left factors of L when we know an LSH S made of $x_i = c_i + \sum_{k=1}^{m} \ell_{ik}\, x_k$ $(1 \leq i \leq m)$ with solution σ such that $\sigma(x_1) = L$. We tackle this problem in two steps: (1) Compute LSHs for all RHLs $L\{h\}^{-1}$ when h ranges over all hedges. (We saw already that this set of RHLs is finite. We call these RHLs the *right derivatives* of L); (2) Use the LSHs produced in step 1 to compute one LSH whose solution contains bindings to all possible intersections of right derivatives of L.

Let $\mathcal{X} = \{x_i \mid 1 \leq i \leq m\}$ and $\mathcal{I} := \{i_1, \dots, i_s\} = \{i \mid x_1 \to^* x_i\}$ where $i_1 = 1$ and $\to^*$ is the reflexive-transitive closure of $\to$ defined by: $x_i \to x_j$ if x_j occurs at horizontal position in the right side of the i-th equation of S.

We have $x_j \doteq_\sigma c_j + \sum_{k=1}^{m} \ell_{jk}\, x_k$ for all $j \in \mathcal{I}$, and if we multiply all these relations to the right with $\{h\}^{-1}$, we obtain $y_j \doteq_\mu d_j(h) + \sum_{k=1}^{m} \ell_{jk}\, y_k$ where $d_j(h) = 1$ if $h \in [\![x_j]\!]_\sigma$ and $d_j(h) = 0$ otherwise, and μ extends σ with $\mu(y_j) := [\![x_j]\!]_\sigma \{h\}^{-1}$ for all $j \in \mathcal{I}$. Hence the LSHs for the right derivatives of L are

$$\begin{array}{ll} y_{i_j} = v_{i_j} + \sum_{k=1}^{s} \ell_{i_j i_k}\, y_{i_k} & (1 \leq j \leq s) \\ x_i = c_i + \sum_{k=1}^{m} \ell_{ik}\, x_k & (1 \leq i \leq m) \end{array} \tag{2}$$

with $(v_{i_1}, \dots, v_{i_s}) \in \Delta := \{(d_{i_1}(h), \dots, d_{i_s}(h)) \mid h \in \mathcal{H}(\Sigma, \emptyset)\}$. In order to compute the set Δ, we define the relation $M \bowtie N$ for $M, N \in 2^{\mathcal{X}}$, with the reading "there exists a hedge h that belongs to $[\![x]\!]_\sigma$ for all $x \in M$ and does not belong to any $[\![x']\!]_\sigma$ when $x' \in N$." Then $(v_{i_1}, \dots, v_{i_s}) \in \Delta$ if and only if there exist $M, N \in 2^{\mathcal{X}}$ such that $M \cup N = \{x_j \mid j \in \mathcal{I}\}$, $M \bowtie N$ holds, and $\{j \in \mathcal{I} \mid v_j = 1\} = \{j \in \mathcal{I} \mid x_j \in M\}$. Thus, in order to compute Δ it is sufficient to be be able to compute the pairs $\langle M, N \rangle \in 2^{\mathcal{X}} \times 2^{\mathcal{X}}$ for which $M \bowtie N$ holds.

It is easy to see that $M \bowtie N$ holds if and only if it can be inferred with

$$\frac{[\forall j \in J.\, c_j = 1 \wedge \forall k \in K.\, c_k = 0]}{\{x_j \mid j \in J\} \bowtie \{x_k \mid k \in K\}}$$

$$\frac{\{x_{u_j} \mid j \in J\} \bowtie \{x_{s_n} \mid n \in N_1\} \quad \{x_{v_j} \mid j \in J\} \bowtie \{x_{t_n} \mid n \in N_2\} \quad [\alpha]}{\{x_j \mid j \in J\} \bowtie \{x_k \mid k \in K\}}$$

where the side condition $[\alpha]$ of the second inference rule is

$$\begin{array}{l} J \cap K = \emptyset \text{ and there is } a \in \Sigma \text{ such that } \forall j \in J.\, a\langle x_{u_j}\rangle\, x_{v_j} \in \mathtt{rs}(x_j) \text{ and} \\ \quad \{a\langle x\rangle\, x' \mid a\langle x\rangle\, x' \in \bigcup_{k\in K} \mathtt{rs}(x_k)\} = \{a\langle x_{s_n}\rangle\, x_{t_n} \mid n \in N\} \text{ and} \\ \quad N_1 \cup N_2 = N \text{ and } N_1 \cap N_2 = \emptyset \end{array}$$

and the meaning of $\mathtt{rs}(x_j)$ is as defined in Sect. 4. The first inference rule is valid because $\epsilon \in \bigcap_{j\in J} [\![x_j]\!]_\sigma \setminus \bigcup_{k\in K} [\![x_k]\!]_\sigma$, whereas the second inference rule is valid because of the existence of a hedge $a\langle h_1\rangle\, h_2 \in \bigcap_{j\in J} [\![x_j]\!]_\sigma \setminus \bigcup_{k\in K} [\![x_k]\!]_\sigma$. These inference rules are finitely branching and they constitute an inductive definition for the relation $M \bowtie N$ defined on a finite set of $2^m \times 2^m$ pairs. Therefore, these inference rules render a decision algorithm for the relation $M \bowtie N$, and this yields an algorithm for the computation of Δ.

We have just seen how to compute LSHs for all right derivatives of L, and that these LSHs share the common structure of (2). Suppose these LSHs are $S_1, \ldots, S_p$ where every S_l is of the form

$$\begin{array}{ll} y^l_{i_j} = v^l_{i_j} + \sum_{k=1}^{s} \ell_{i_j i_k}\, y^l_{i_k} & (1 \le j \le s) \\ x_i = c_i + \sum_{k=1}^{m} \ell_{ik}\, x_k & (1 \le i \le m) \end{array}$$

with $i_1 = 1$ and the set of variables $\{y^l_{i_1}, \ldots, y^l_{i_s}\}$ besides the set of variables $\mathcal{X}$ that is shared by all of them. Note that $p = |\Delta| \le 2^s$ because $\Delta \subseteq \{0,1\}^s$. Let's denote the unique solution of S_l by σ_l.

The left factors of L are the elements of the set $\left\{\bigcap_{l\in G} [\![y^l_1]\!]_{\sigma_l} \mid G \in 2^{\{1,\ldots,p\}}\right\}$. We consider the set of variables $\mathcal{Z} := \left\{\bigcap_{l\in G} y^l_{k_l} \;\middle|\; G \in 2^{\{1,\ldots,p\}} \wedge \forall l \in G.\, k_l \in \mathcal{I}\right\} \cup \left\{\bigcap_{i\in H} x_i \mid H \in 2^{\{1,\ldots,m\}}\right\}$ where variable names are identified modulo associativity, commutativity, and idempotency of intersection. We will construct an LSH LF with variables from $\mathcal{Z}$ whose unique solution μ satisfies the conditions: (c_1) $\mu(\bigcap_{l\in G} y^l_{k_l}) = \bigcap_{l\in G} [\![y^l_{k_l}]\!]_{\sigma_l}$ for every variable $\bigcap_{l\in G} y^l_{k_l}$ that occurs in LF; and (c_2) $\mu(\bigcap_{i\in H} x_i) = \bigcap_{i\in H} [\![x_i]\!]_\sigma$ for every variable $\bigcap_{i\in H} x_i$ that occurs in LF. Our main requirement is that variables of $\{\bigcap_{l\in G} y^l_1 \mid G \subseteq \{1,\ldots,p\}\}$ appear in LF. Then LF can be regarded as LSH for every left factor of L because every left factor of L is $[\![\bigcap_{l\in G} y^l_1]\!]_\mu$ for some $G \in 2^{\{1,\ldots,p\}}$, and a rearrangement of the equations of LF which places the equation for $\bigcap_{l\in G} y^l_1$ first is an LSH for the left factor $[\![\bigcap_{l\in G} y^l_1]\!]_\mu$. LF is constructed incrementally, by intersecting equations of the LSHs $S_1, \ldots, S_p$:

- For every $G \in 2^{\{1,\ldots,p\}}$ we intersect the first equations of the LSHs from the set $\{S_l \mid l \in G\}$. The intersection of any number of equations is the obvious generalization of the intersection operation of 2 equations described

in Sect. 3. There are 2^p such intersections, and they will produce 2^p equations with variables $\bigcap_{l\in G} y_1^l$ in their left sides. We add these equations to LF.
- For every variable $\bigcap_{l\in G} y_{k_l}^l$ that occurs at horizontal position in some equation of LF, add (if missing) to LF the equation obtained by intersecting the equations of the set $\{k_l$-th equation of $S_l \mid l \in G\}$.
- For every variable $\bigcap_{i\in H} x_i$ that occurs in some equation of LF, add (if missing) to LF the equation produced by intersecting the equations of the set $\{i$-th equation of the LSH for $L \mid i \in H\}$.

This process terminates because $\mathcal{Z}$ is finite, so we can not add indefinitely equations to LF. We end up with LF being an LSH with properties (c_1) and (c_2).

6 Conclusion

LSHs are a representation of RHLs that is suitable for performing several operations that show up in the analysis and processing of XML. The algorithms described here indicate how the intersections, quotients, and the left and right factors of RHLs can be computed when using the LSH formalism. It should be mentioned, however, that there are also several operations for which LSHs are not a suitable representation, such as the computation of symmetric language.

References

1. Antimirov, V.M.: Partial derivatives of regular expressions and finite automaton constructions. Theoretical Computer Science 155, 291–319 (1996)
2. Conway, J.H.: Regular Algebra and Finite Machines. Mathematics series. Chapman and Hall, Boca Raton (1971)
3. Hosoya, H., Vouillon, J., Pierce, B.C.: Regular expression types for XML. ACM Transactions on Programming Languages and Systems 27(1), 46–90 (2005)
4. Kozen, D.C.: Automata and Computability. Undergraduate Texts in Computer Science. Springer, New York (1997)
5. Marin, M., Kutsia, T.: Computational methods in an algebra of regular hedge expressions. RISC Report Series 09-03, RISC-Linz (March 2009)
6. Murata, M.: Hedge automata: a formal model for XML schemata (1999), http://www.xml.gr.jp/relax/hedge_nice.html
7. Murata, M.: Extended path expressions for XML. In: Proceedings of the 20th symposium on Principles of Database Systems (PODS 2001), Santa Barbara, California, USA, pp. 126–137. ACM, New York (2001)
8. Murata, M., Lee, D., Mani, M., Kawaguchi, K.: Taxonomy of XML schema languages using formal language theory. ACM Transactions on Internet Technology 5(4), 660–704 (2005)

Element Algebra

Manuk G. Manukyan

Yerevan State University
Yerevan, 0025
mgm@ysu.am

Abstract. An element algebra supporting the element calculus is proposed. The input and output of our algebra are *xdm-elements*. Formal definition of element algebra is offered. We consider algebraic expressions as mappings. A reduction of the element calculus to the element algebra is suggested.

1 Introduction

The XML databases currently attract definite interest among researchers of databases for the following reasons:

- DTD is a compromise between the strict-schema models such as the relational or object models and the completely schemaless world of semi-structured data;
- in contrast to semi-structured data model, the concept of database schema in the sense of conventional data models is supported;
- in contrast to conventional data models strict-schemas, there is possibility to define more flexible database schemas (DTDs often allow optional fields or missing fields, for instance) [12].

A big disadvantage of DTD is that it does not contain tools to include information of types and integrity constraints. An important step in this direction is the XML Schema [3,18] which is a formalism to restrict the structure of XML documents and also to extend XML with data types. An XML query data model [10] is developed which is based on the XML Schema type system. Notice that the XML query model is the foundation of the XML query algebra [9]. In the context of XML query data model and XML query algebra an XML query language [4] is suggested. Notice that XML Schema = XML + data types. Here data type has a non-classical interpretation: The value set of data type is defined without corresponding operations [8]. Therefore on the level of XML Schema we can not define the dynamics of application domain objects. XQuery [4] is a query language for XML which allows to give queries across all these kinds of data, whether physically stored in XML or viewed as XML via middleware. XQuery is not a declarative query language (detailed see in [8]). In distinct to declarative languages it is impossible to create an effective optimizer for XQuery due to its procedural character. In [16] we suggested an extensible data model (xdm) to:

- extend semantics of the XML data model for supporting database concept;
- create a declarative query language.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 113–120, 2010.

It is common in database theory to translate a query language into an algebra since algebra is a language of execution level. Thus the algebra is used to:

-give a semantics for the query language;

-support query optimization.

The requirements above presume formal definition of the algebra. We have developed an element algebra to support the element calculus (a declarative query language for xdm) [16]. The input and output of our algebra are xdm-elements. The considered algebra supports standard algebraic operations. In the case of standard algebra the operands of algebraic operations are relations. In our case the operands of algebraic operations are the xdm-elements. Thus to directly apply the standard algebraic operations to xdm-elements we needs:

- formalization the xdm-element in compliance with the theory of relational databases;
- defining the inference rules of the resulting schemas of algebraic expressions;
- proving element calculus and element algebra equivalence.

2 Related Work

Many XML algebras are considered in literature, for example [1,2,5,11,13,14,17,20,21]. Some XML algebras considered in [1,11,13,14] have been developed to support XQuery. In fact, the XML data model could be either a tree or a graph. A forest could be transformed to a single tree by simply adding a root node as a common parent for all trees. The basic unit of information is an individual member of a collection feeding operators. Notice that operator takes relations as input and produces a relation as output in the relational algebra. The relation composes of tuples which are basic units of information in the relational algebra. In [21] an XML algebra (called XAL) for data mining has been offered. In XAL, each XML document is represented as a rooted directed graph with a partial order relation defined on its edges. The basic unit is a vertex representing either element or attribute. An operator receives set of vertices as input and produces set of vertices as output. XAL provides a set of equivalence rules. Based on these rules a heuristic algorithm to transform a query tree into on optimized tree has been suggested. In Niagara [20] the XML document is also represented as rooted directed graph with elements and attributes as vertrices. The basic unit is a bag of vertices. Thus the operators operate with collections of bags of vertices. This approach to XML algebra assumes an implementation independent optimization by rewriting using equivalence. TAX [13,14] treates an XML document as a forest of labeled rooted trees. TAX takes a labeled rooted tree as a basic unit by introducing the notation of pattern tree and witness tree. A pattern tree ia a pair of $P = (T, E)$, where T is a node-labeled and edge-labeled tree, E is a formula with value-based predicates applicable to tree nodes. Each node in T is labeled by a unique integer whereas each edge T is labeled by either pc (parent-child) or ad (ancestor-descendant). A witness tree is an instance of the data trees matching the pattern tree. All operations of this algebra take collections of trees as input and produce a collection of trees as

output. In [1] an XML algebra (called IBM) to support XQuery is considered. The basic unit is a vertex that represents either element, attribute, or reference. An operator receives a collection of vertices as input and produces a collection of vertices as output. This model is a logical model and nothing is specified about the underlying storage representation or physical operators. In addition to standard operations a new reshaping operation to create a new XML document from fragments of selected XML documents is offered. A YATL [5] algebra has been developed for an XML-based integration system which integrates data from different sources. Only two new operations the *bind* and the *Tree* are suggested. The *bind* operation is used to extract relevant information from different sources and produce a structure called *Tab*, which practically is a 1NF relation [7]. The *Tree* is the inverse operation to *bind* and generates a new XML document. All others are standard operations of relational algebra. An algebra is considered in [17] for a DBMS designed specifically for managing semi-structured data. The distinguishing feature of this approach is on cost-based query optimization and manipulating dynamic data structures. Each query is transformed into a logical query plan using logical operations such as *select*, *project*, *name*, etc. which can be considered algebra operations. A cost-based approach is used to select the best physical plan from generated physical plans. Another XML algebra, called AT&T, is considered in [11]. The AT&T algebra is powerful enough to capture the semantics of many XML query languages and several optimization rules have been specified. In this algebra most of the operations are based on the iteration operation. AT&T has distinctive ability in detecting errors at query compile time with its well-defined list of the operations. A tree based algebra (called TA) is considered in [2]. The basic unit is a tree that is used to model an XML data. In this algebra operations take trees as input and produce tree as output. While the IBM, Niagara, TAX, XAL, AT&T, TA algebras were proposed as standalone XML algebras, the Lore [17] and YATL were developed for the semi-structured database system and integration system, respectively. The Niagara, TAX, XAL, AT&T, TA algebras support standard algebraic operations.

3 Formal Definition of Element Algebra

Definition 1. *We say that S is an xdm-element schema, if*
1. S=<name, atomictype, f>, where $f \in \{?, *, +, \perp$ [1]$\}$, *or*
2. S=<name, typeOp(S_1*,* S_2*, ... ,* S_n*), f>,* $typeOp \in \{sequence, choice, all\}$, *and* S_i *is an xdm-element schema*[2], $1 \leq i \leq n$.

Definition 2. *The xdm-element s of schema S is a finite collection of mappings* $S \rightarrow domain(firstComp(S)) \times domain(secondComp(S))$*; if secondComp(S)= typeOp(*S_1*,* S_2*, ... ,* S_n*) then the following constraint should be hold for all* $e \in s$*:* $e[S_i] \in domain(S_i)$, $1 \leq i \leq n$.

[1] A $\perp$ following an xdm-element means that the xdm-element may occur exactly one time.
[2] The xdm-attributes are not considered for simplicity.

The $firstComp, secondComp, domain$ functions have an obvious semantics in the previous definition. Notice that

$$domain(secondComp(S)) = \begin{cases} valSet(atomictype), & \text{if } secondComp(S) = \\ & \text{atomictype} \\ \bigcup_{i=1}^{n} domain(S_i), & \text{if not} \end{cases}$$

Definition 3. *Let R and Q be xdm-elements schemas. We say that R and Q are similar, if*
1. secondComp(R)=atomictype1, secondComp(Q)=atomictype2, and atomictype1=atomictype2, or
2. secondComp(R)=typeOp(R_1, R_2, ... , R_n), secondComp(Q)=typeOp(Q_1, Q_2, ... , Q_n), and R_i, Q_i are similar, $1 \leq i \leq n$.

Definition 4. *Let R and Q be xdm-elements schemas. We say that R is subschema of Q ($R \subseteq Q$), if*
1. firstComp(R)=name1, secondComp(R)=atomictype1, firstComp(Q)=name2, secondComp(Q)=atomictype2, and name1=name2, atomictype1=atomictype2, or
2. firstComp(R)=name1, secondComp(R)=typeOp(R_1, R_2, ... , R_k), firstComp(Q)=name2, secondComp(Q)=typeOp(Q_1, Q_2, ..., Q_m), and name1=name2, and $\forall i \in [1,k]\ \exists j \in [1,m]$ that $R_i \subseteq Q_j$.

Definition 5. *Let r and q be xdm-elements with R and Q similar schemas correspondingly*[3]*. Let us say that r and q are equal, if*
1. secondComp(R)=secondComp(Q)=atomictype, and content(r)=content(q), or
2. secondComp(R)=typeOp(R_1, R_2, ... , R_n), secondComp(Q)=typeOp(Q_1, Q_2, ... , Q_n):
a) typeOp=sequence, $\forall i \in [1,n]$ firstComp(R_i)=firstComp(Q_i), and r_i and q_i are equal xdm-elments with similar schemas R_i and Q_i correspondingly;
b) typeOp=all, $\forall i \in [1,n]\ \exists j \in [1,n]$ firstComp(R_i)=firstComp(Q_j), and r_i and q_j are equal xdm-elments with similar schemas R_i and Q_j correspondingly;
c) typeOp=choice, there is a unique $i \in [1,n]$ such that the following holds for some unique $j \in [1,n]$: firstComp(R_i)=firstComp(Q_j), and r_i and q_j are equal xdm-elments with similar schemas R_i and Q_j correspondingly.

Concantenation. The concantenation of xdm-elements $r =< namer, r_1, r_2, \ldots, r_k >$ and $q =< nameq, q_1, q_2, \ldots, q_m >$ is an xdm-element defined as follows:

$\widehat{rq} =< \widehat{rq}, r_1, r_2, \ldots, r_k, q_1, q_2, \ldots, q_m >$

Set-theoretic operations. In definition of set-theoretic operations union, intersection and difference it is assumed that schemas of operands are similar. Let r and q be xdm-elements with R and Q similar schemas correspondingly[4]. The

[3] Without loss of generality it is assumed that an xdm-element schema is a pair of the following type <name, type>.
[4] We will use $\langle\rangle$ to signify a multiset, {} to denote a set, while [] symbolizes a list.

union, intersection and difference of r and q xdm-elements are the xdm-elements defined as follows:

$r \cup q =< r \cup q, \langle t | t \in r \vee t \in q \rangle >$

$r \cap q =< r \cap q, \langle t | t \in r \wedge t \in q \rangle >$

$r - q =< r - q, \langle t | t \in r \wedge t \notin q \rangle >$

Notice that union and intersection are commutative and associative operations. *Cartesian Product.* Let r and q be the xdm-elements with R and Q schemas correspondingly. The Cartesian product of r and q xdm-elements is an xdm-element defined as follows:

$r \times q =< r \times q, \langle \widehat{ts} | t \in r \wedge s \in q \rangle >$.

Selection. Let r be an xdm-element with schema R, and P be a predicate. The result of operation of selection from r by P is an xdm-element defined as follows:

$\sigma_P(r) =< \sigma_P(r), \langle t | t \in r \wedge P(t) \rangle >$

Projection. Let r be an xdm-element with schema R and $\pi_L(r)$ be a projection operation, where L is a list of elements. For simplicity let us assume $L = [A, E \rightarrow Z, X \rightarrow Y]$ $(A, X \in R)$, then the result of the projection operation is an xdm-element defined as follows:

$\pi_L(r) =< \pi_L(r), \langle < name, \widehat{t[A]Zt[Y]} > | t \in r \wedge Y = X \wedge Z := E \rangle >$

Natural Joins. Let r and q be xdm-elements with R and Q schemas correspondingly, such that $R \not\subseteq Q$ and $Q \not\subseteq R$ and $R \cap Q \neq \emptyset$. The natural join of r and q xdm-elements is an xdm-element defined as follows:

$r \bowtie q =< r \bowtie q, \langle \widehat{ts[\bar{L}]} | t \in r \wedge s \in q \wedge t[L] = s[L] \rangle >$, where $L = R \cap Q, \bar{L} = Q - L$

Grouping. Let r be an xdm-element with schema R and $\gamma_L(r)$ be a grouping operation, where L is a list of elements. For simplicity let us assume $L = [A, f(B) \rightarrow C]$ $(A, B \in R, f \in \{min, max, sum, count, average\})$, then the result of the grouping operation is an xdm-element defined as follows:

$\gamma_L(r) =< \gamma_L(r), \{\widehat{t[A]s} | t \in r \wedge s =< C, f(\pi_B(\sigma_{A=t[A]}(r))) >\} >$

Notice that our algebra also includes the conventional *theta joins, duplicate elimination, division, renaming, sorting* operations and aggregate functions.

4 Algebraic Expressions as Mappings

We will use Exp and $schema(Exp)$ to signify an algebraic expression and its schema correspondingly. Let us define the following operations $\oplus$, $\otimes$ and $\ominus$:

$?\oplus? =?$	$?\otimes? =?$	$?\ominus? =?$	$+\ominus \perp = *$	$*\ominus \perp = *$	$\perp \otimes \perp =?$
$? \oplus * = *$	$? \otimes * =?$	$? \ominus * =?$	$\perp \ominus \perp =?$	$+ \oplus + = +$	$* \otimes + = *$
$? \oplus + = *$	$? \otimes + =?$	$*\ominus? = *$	$* \ominus + = *$	$+ \otimes + = *$	$\perp \ominus? =?$
$?\oplus \perp =?$	$?\otimes \perp =?$	$? \ominus + =?$	$+ \ominus * = *$	$\perp \ominus + =?$	$\perp \ominus * =?$
$\perp \oplus * = *$	$\perp \otimes * =?$	$+\ominus? = *$	$* \ominus * = *$	$\perp \oplus \perp = \perp$	$* \oplus * = *$
$\perp \oplus + = +$	$\perp \otimes + =?$	$?\ominus \perp =?$	$+ \ominus + = *$	$* \oplus + = *$	$* \otimes * = *$

The following recursive rules are used to define $schema(Exp)$:

$r1$. If $Exp = r$, where r is an xdm-element with schema R, then $schema(Exp) =< Exp, secondComp(R), thirdComp(R) >$;

$r2$. If $Exp = Exp_1 \cup Exp_2$ or $Exp = Exp_1 \cap Exp_2$, or $Exp = Exp_1 - Exp_2$, then if

a) $secondComp(schema(Exp_1)) = secondComp(schema(Exp_2)) = atomictype$, and $schema(Exp) =< Exp, secondComp(schema(Exp_1)), thirdComp(schema(Exp_1))\ Op\ thirdComp(schema(Exp_2)) >$, where

$$Op = \begin{cases} \oplus, \text{ if } Exp = Exp_1 \cup Exp_2 \\ \otimes, \text{ if } Exp = Exp_1 \cap Exp_2 \\ \ominus, \text{ if } Exp = Exp_1 - Exp_2 \end{cases}$$

b) $secondComp(Exp_1) = typeOp(schema({Exp^1}_1), schema({Exp^2}_1), ..., schema({Exp^n}_1))$, $secondComp(schema(Exp_2)) = typeOp(schema({Exp^1}_2), schema({Exp^2}_2), ..., schema({Exp^n}_2))$, and $schema(Exp) =< Exp, typeOp(schema({Exp^1}_3), schema({Exp^2}_3), ..., schema({Exp^n}_3)), thirdComp(schema(Exp_1))\ Op\ thirdComp(schema(Exp_2)) >$, where $\forall i \in [1, n]\ schema({Exp^i}_3) =< {Exp^i}_3, secondComp(schema({Exp^i}_1)), thirdComp(schema({Exp^i}_1))\ Op\ thirdComp(schema({Exp^i}_2))$;

$r3$. If $Exp = Exp_1 \times Exp_2$, then $schema(Exp) =< Exp, sequence(secondComp(schema(Exp_1)), secondComp(schema(Exp_2))), thirdComp(schema(Exp_1)) \oplus thirdComp(schema(Exp_2)) >$;

$r4$. If $Exp = \sigma_P(Exp_1)$, then $schema(Exp) =< Exp, secondComp(schema(Exp_1)), thirdComp(schema(Exp_1)) \otimes " * " >$;

$r5$. If $Exp = \pi_L(Exp_1)$ (in general case $L = sequence(L_1, L_2, L_3)$, where $L_1 \subseteq secondComp(schema(Exp_1))$, L_2 is a list of renamed xdm-elements, and L_3 is a list of derived xdm-elements), then $schema(Exp) =< Exp, L, thirdComp(schema(Exp_1)) >$;

$r6$. If $Exp = Exp_1 \bowtie Exp_2$, then $schema(Exp) =< Exp, secondComp(schema(Exp_1)) \cup secondComp(schema(Exp_2)), (thirdComp(schema(Exp_1)) \oplus thirdComp(schema(Exp_2))) \otimes " * " >$;

If in Exp the $r_1, r_2, ..., r_n$ xdm-elements with $R_1, R_2, ..., R_n$ schemas are used, then Exp is defined by the following mapping:

$Exp : Coll(R_1) \times Coll(R_2) \times ... \times Coll(R_n) \to Coll(schema(Exp))$, where $Coll(R)$ is a collection of all xdm-elements with schema R.

5 Element Calculus Reduction to Element Algebra

An expression in the element calculus has the following type[5]: $\langle x_1 x_2 ... x_k | \psi(x_1, x_2, ..., x_k) \rangle$, where ψ is a formula with $x_1, x_2, ..., x_k$ free variables. Let s be an xdm-element of schema S and $E(S)$ is defined as follows:

$E(S) = \pi_1(s) \cup \pi_2(s) \cup ... \cup \pi_n(s)$. If $s_1, s_2, ..., s_n$ occur in ψ, then $DOM(\psi) = E(S_1) \cup E(S_2) \cup ... \cup E(S_n) \cup \{\alpha_1, \alpha_2, ..., \alpha_n\}$, where $\forall i \in [1, n]\ \alpha_i$

[5] This section is based on the similar facts and techniques of the theory of relational databases [15,19].

is a constant of ψ. Let E be an expression of element algebra defined as follows: $E : DOM(\psi) \rightarrow DOM(\psi)$. Let us proof that for each *safety* expression of element calculus an equivalent expression of element algebra exists. For that, for each subformula ω of ψ we recursively define the expression of element algebra equivalent to $\langle y_1 y_2 ... y_m | \omega(y_1, y_2, ..., y_m) \rangle$. Notice that from *safety* of expression of element calculus, does not foolow *safety* of subformulas of the formula. Therefore we search for the equivalent expression of element algebra for $(DOM(\psi))^m \cap \langle y_1 y_2 ... y_m | \omega(y_1, y_2, ..., y_m) \rangle$, where ω is a subformula of ψ and D^m means $D \times D \times ... \times D$ (m times).

Case 1. If subformula ω is an atom of type $x\theta y$, $x\theta x$, $x\theta c$, $c\theta x$ (where x and y are element calculus variables, c is a constant, and $\theta \in \{=, \neq, >, \geq, <, \leq\}$), then $\sigma_{x\theta y}(E \times E)$, $\sigma_{x\theta x}(E)$, $\sigma_{x\theta c}(E)$, and $\sigma_{c\theta x}(E)$ are equivalent expressions of element algebra correspondingly.

Case 2. The subformula ω is an atom of type $(pe)(x_1, x_2, ..., x_l)$, where $\forall i \in [1, l]$ x_i is element calculus variable, and (pe) is the result of path expression [6] pe converted to multiset. Notice that the path expression is a sequence of steps defined as follows: $[/|//]Step_1[/|//]Step_2[/|//]...[/|//]Step_n$. The equivalent expression of element algebra for the path expression is created by the following recurrent relation:

$$Step_{i+1} = \begin{cases} \pi_{L_i}(Step_i), & \text{if condition is not given} \\ \pi_{L_i}(\sigma_P(Step_i)), & \text{if P is a predicate} \\ \pi_{L_i}(\sigma_P(\gamma_{L_1}(Step_i))), & \text{if condition is given by a aggregate function} \end{cases}$$

here $i = 0, 1, ..., n-1$, $Step_0$ = initial xdm-element, L_i is the resulting list both of the xdm-elements and attributes in the $step_{i+1}$, L_1 = grouping xdm-element/attribute + aggregate function $\rightarrow$ xdm-element. If ae is the equivalent expression of our algebra for the path expression, then $\pi_L(ae)$ will be an equivalent expression of element algebra for $(pe)(x_1, x_2, ..., x_l)$, where $L = [x_1, x_2, ..., x_l]$.

Case 3. $\omega(y_1, y_2, ..., y_m) = \neg\omega_1(y_1, y_2, ..., y_m)$. If E_1 is equivalent expression of element algebra for $(DOM(\psi))^m \cap \langle y_1 y_2 ... y_m | \omega_1(y_1, y_2, ..., y_m) \rangle$, then $E^m - E_1$ is an equivalent expression of element algebra for $(DOM(\psi))^m - \langle y_1 y_2 ... y_m | \omega_1(y_1, y_2, ..., y_m) \rangle$, which is equivalent to $(DOM(\psi))^m \cap \langle y_1 y_2 ... y_m | \neg\omega_1(y_1, y_2, ..., y_m) \rangle$.

Case 4. $\omega(y_1, y_2, ..., y_m) = \omega_1(u_1, u_2, ..., u_n) \vee \omega_2(v_1, v_2, ..., v_l)$. Let E_ω be equivalent expression element algebra for $(DOM(\psi))^m \cap \langle y_1 y_2 ... y_m | \omega(y_1, y_2, ..., y_m) \rangle$. If E_{ω_1} and E_{ω_2} are equivalent expressions element algebra for $(DOM(\psi))^n \cap \langle u_1 u_2 ... u_n | \omega_1(u_1, u_2, ..., u_n) \rangle$ and $(DOM(\psi))^l \cap \langle v_1 v_2 ... v_l | \omega_2(v_1, v_2, ..., v_l) \rangle$ correspondingly, then $E_\omega = \pi_{y_1, y_2, ..., y_m}(E_{\omega_1} \times E^{m-n}) \cup \pi_{y_1, y_2, ..., y_m}(E_{\omega_2} \times E^{m-l})$.

Case 5. $\omega(y_1, y_2, ..., y_m) = (\exists y_{m+1})\omega_1(y_1, y_2, ..., y_{m+1})$. Let E_1 be equivalent expression of element algebra for $(DOM(\psi))^{m+1} \cap \langle y_1 y_2 ... y_{m+1} | \omega_1(y_1, y_2, ..., y_{m+1}) \rangle$. It is obvious that $y_{m+1} \in DOM(\psi)$ as ψ is a *safety* formula.

Thus $\pi_{y_1, y_2, ..., y_m}(E_1)$ is equivalent expression of element algebra for $(DOM(\psi))^m \cap \langle y_1 y_2 ... y_m | (\exists y_{m+1})\omega_1(y_1, y_2, ..., y_{m+1}) \rangle$.

It is easy to see that we can analogously create the equivalent algebraic expressions for have not considered formulas.

6 Conclusion

An element algebra supporting the element calculus is proposed. The xdm-elements are inputs and outputs for the suggested algebra. Formal definitions of xdm-element schema, xdm-element with given schema, similar schemas, subschemas and equal xdm-elements are given. Based on these definitions an element algebra is formally defined. The equivalence rules for algebraic expressions are presented. The algebraic expressions are considered as mappings. Inference rules of resulting schemas of algebraic expressions are offered. The equivalence of element calculus and element algebra is proved. Finally, our approach to XML algebra allows to apply relational optimization techniques.

References

1. Beech, D., Malhotra, A., Rys, M.: A formal data model and algebra for XML. Communication W3C (1999)
2. El Bekai, A., Rossiter, N.: A tree based algebra framework for XML data systems. In: ICEIS (2005)
3. Biron, P., Malhotra, A.: XML Schema Part2: Datatypes (2001), http://www.w3.org
4. Chamberlin, D., Florescu, D., Robie, J., Simeon, J., Stefanescu, M.: XQuery: A Query Language for XML (2001), http://www.w3.org
5. Christophides, V., Cluet, S., Simeon, J.: On wrapping, query languages and efficient XML integration. In: ACM SIGMOD Conference on Management of Data (2000)
6. Clark, J., DeRose, S.: XML Path Language (1999), http://www.w3.org
7. Codd, E.: A relational model for large shared data banks. Commuincations of the ACM (1970)
8. Date, C.: An Introduction to Database Systems. Addison-Wesley, Reading (2004)
9. Fankhauser, P., et al.: The XML Query Algebra (2001), http://www.w3.org
10. Fernandez, M., Robie, J.: XML Query Data Model (2001), http://www.w3.org
11. Fernandez, M., Simeon, J., Walder, P.: A semi-monad for semi-structured data. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, p. 263. Springer, Heidelberg (2000)
12. Garcia-Molina, H., Ullman, J., Widom, J.: Database Systems: The Complete Book. Prentice-Hall, Englewood Cliffs (2002)
13. Jagadish, H., et al.: Tax: A tree algebra for XML. In: DBLP Conference (2001)
14. Jagadish, H., et al.: Timbler: A native XML database. In: VLDB (2002)
15. Maier, D.: The Theory of Relational Databases. Computer Science Press, Rockville (1983)
16. Manukyan, M.: Extensible data model. In: ADBIS (2008)
17. McHugh, J., et al.: Lore: A database management system for semi-structured data. In: SIGMOD (1997)
18. Thompson, H., Beech, D., et al.: XML Schema Part1: Structures, http://www.w3.org
19. Ullman, J.: Principles of Database Systems. Computer Science Press, Rockville (1980)
20. Viglas, S., et al.: Putting XML Query Algebras into Context (2002), http://www.cs.wisc.edu/niagara/Publications.html
21. Zhang, M., Yao, J.: XML algebra for data mining. In: SPIE (2004)

Shall I Trust a Recommendation? Towards an Evaluation of the Trustworthiness of Recommender Sites

G. Lenzini, Y. van Houten, W. Huijsen, and M. Melenhorst

Novay, Brouwerijstraat 1
NL-7523 XC – Enschede, The Netherlands

Abstract. We present a preliminary study the aim of which is to provide a high level model for the evaluation of the trustworthiness of recommender systems as e-commerce services. We identify and comment on relevant trustworthiness indicators for the following different perspectives: user interface, content information and quality of recommendations.

Keywords: Trustworthiness, Recommender Systems, Review sites.

1 Introduction

Consumers book their holidays on Tripadvisor (www.tripadvisor.com), on Amazon (www.amazon.com) they receive recommendations for products they might also want to buy, and people share their tastes, experiences and suggestions on Kaboodle (www.kaboodle.com). These examples show that more and more online shopping activities are becoming a social process, just as it often happens in a regular shopping mall. Moreover, there is a growing need for information re-education and classification, as the abundance of reviews, suggestions and products is overwhelming to consumers. Hence, recommender systems are introduced to help users cope with the options they can choose from. As shopping in the real world involves trusting advices and recommendations from other people, trust is also an important factor that must be considering when designing an online shopping service. In this paper we address trust factors within e-commerce recommender sites.

Despite being welcomed with some reserve years ago, the concept of trust in digital information is nowadays widely accepted. Only few years ago trust was considered possible only among people, but successive studies in human-computer interaction have indicated that people can have relations (including the social relation of trust) also with computer technology. According to [10] "*individuals can be induced to behave* as if *computers warranted human treatment, even though users* know *that the machines do not actually warrant this treatment*". In other words, modern computer technology has the ability to influence human perception so that computers can be perceived *as if* they were human partners [2]. Therefore, comprehensive models of trust, originally designed to be applied in social science, have been adapted to fit digital information. Trust models can now describe trust in digital information as well as

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 121–128, 2010.

trust in information systems, in e-commerce, and in on-line relationships [6]. Trust is thus the intervening variable that directly affects the use of a certain technology [10]. This statement is actually even truer for web sites, which today offer their users a wide range of features, some of these explicitly pushing social aspects.

This paper focuses on recommender sites within the e-commerce domain. Our research's goal is to compose a criterion for the evaluation of the trustworthiness of e-commerce and recommender sites. Part of this evaluation can be done automatically, yielding trustability indicators that may support a higher-level trust evaluation. In its simpler version, this criterion is a checklist that arranges the relevant aspects that a recommender site has to possess to be considered trustworthy. In its most advanced version, it can lead to the development of a tool that automatically evaluates a web site and gives back the site's trustworthiness rank. We underline that our approach to trust does not target the improvement of the accuracy of reviews and recommendations as it happens; we are not interested in this computational approach to trust. Instead, we look at a recommender site as an e-commerce service, and when we talk about the trustworthiness of a recommender site we mean the perception of trust that a user has when using that service. We aim to improve the understanding of the aspects that influence the perception of trust that a user has when choosing a certain recommender site instead of another. From this point of view, the accuracy of the predictions that a recommender system is able to provide, is only one of the possible variables that can affect the overall trustworthiness of the service.

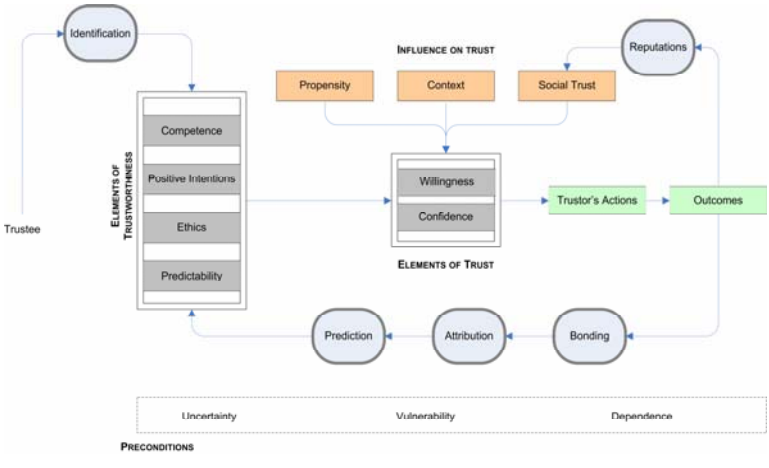


Fig. 1. Trust model (adapted from [6]). At the bottom of the picture there are the preconditions of trust; in double squares, the elements of trustworthiness and the elements of trust; in the squares at the top of the figure, there are the factors that influence trust; in round boxes, the processes of trust.

2 Trust in Recommender Sites

Among the different natures of trust (individual, interpersonal, relational, societal) we are interested in the interpersonal trust. "Interpersonal" means that trust is treated as a

tie between two entities, the *trustor* and the *trustee*. This tie is defined in terms of an attitude that the trustor holds towards the trustee. Among the existing trust models we have selected that by Kelton *et al* [6]. It has been presented as a trust model for digital information and we consider it a candidate to describe trust in recommender sites. The main elements on this model are depicted in Figure 1; the reader can refer to [6 for a detailed description of them.

In the following section we instantiate the Kelton's model to describe trust in our objective of research: the recommender sites. First we observe that the *preconditions of trust* are fulfilled in the specific context of recommender sites. Because there is no standard procedure that helps users evaluate the quality of information, users are forced to work in a situation of uncertainty; users may suffer from potential harm when relying upon reviews and recommendations: A misinformation may bring to a loss of time, to a loss of money, or both. Therefore users find themselves in a position of dependence when resorting to a recommender site and to the information it provides, and they have to balance the risk of a loss and the probability of a gain before taking a decision to trust and proceeding with a purchase. Users that have confidence in the proper functioning and in the reliability of a recommender site have, nevertheless, the possibility of refusing a recommendation, of quitting the process of purchasing, and of switching to a different site. In some cases, even after having paid, users have the option to return the good back to the seller and to be refund of what paid. The *elements of trust* (i.e., confidence and willingness) are then well specified in the context of recommender sites.

In the following sections we provide an overview of the variables that, according to the literature, are related with the elements of trustworthiness of a recommendation service. Where possible, we also indicate how they support and influence the processes of establishing trust. Our analysis addresses *user interface*, *linguistic content* and *quality of recommendations* as relevant aspects for a recommender site. As far as possible, all the elements we are going to analyze are related to the model presented in Figure 1.

2.1 User Interface Analysis

User interfaces are the users' gateways to services. In fact, elements of trustworthiness emerge, and are influential, already at this level: For instance, a badly designed interface may be perceived as a sign of lack of professionalism. Users' willingness to engage with a website is strongly dependent on the extent to which the site succeeds in conveying trustworthiness. Hertzum *et al.* [4] indicate that trust emphasizes that even seemingly objective factors are, actually, perceived factors. Trust does not reside in sources as a label that can be read, but has to be established actively by the individual person: it is ultimately the information seeker's responsibility to assess the source trustworthiness. Thus, websites – including recommender sites - must be accessible in ways that enable information seekers to form an opinion about their trustworthiness. The available elements in a user interface thus influence the subjective perception of trust.

Shneiderman [12] provides a number of guidelines for ensuring users that they are engaging in a trust relationship. First, a website should disclose patterns of past performance. For example, airlines should report on the percentages of flights that landed

without delays. Information about the organization and its management, employees and history may also engage and assure skeptical users. Second, a website should provide references from past and current users. Opinions and experiences of other users provide valuable information on a product or a service. Third, a website should get certifications from third parties. For example, seals of approval from consumers and professional groups (e.g. displayed in the form of logos) help to establish trust. Of course, means for verifying the reliability of the information provided should also be present. Fourth, it should be easy to locate, read, and enforce policies involving privacy and security: Privacy policies that are difficult to find and that are incomprehensible or awkward to read only undermine trust.

Of course, users may question whether all this information eliciting trust is actually trustworthy information. What if comments of satisfied costumers were actually written from employees of that company? One way to improve the users' perception of trust is to allow users to compare information, for example by allowing users to have easy access to different sources of information and to check whether the information is consistent. The system the user interacts with should facilitate making these kinds of comparisons. In the case of recommendations, multiple recommendations should be available, and the source of the recommendations should be explicitly visible. When applied to recommender sites, first-hand experiences are related to a person's own experience with the trustworthiness of the system as a whole, or of a reviewer in particular. Reputations of reviewers can be managed by the systems, e.g., by granting top reviewers a visible top status and their characterizing attributes. Surface attributes refer to user interface elements, including the name and use of words by the reviewer. General assumptions and stereotypes concern expectations about the interests and the competence of a reviewer. In sum, the user interface elements described in [12] can not only be applied to the recommender system as a whole, but also to reviewers providing a recommendation. All these aspects are related to the elements of trustworthiness as presented in the model displayed in Figure 1: *competence,* positive *intention,* ethics*,* and *predictability.*

2.2 Automatic Trustability Analysis of Content

The trustability of an individual or of a piece of content (e.g., a written review) can be gauged from a number of indicators, such as the text's internal consistency, and the author's reputation. The indicators can be considered input to the trustor; this process can be supported programmatically by analyzing the content of the trustee. Trustability indicators are measures of aspects of the extent to which a text can be trusted, and provide a rough sense of trustability: a text may score poor on an indicator and still be trustworthy. It is by an overview of a number of indicators that a general trustability may be determined. In the following we discuss some of these indicators, and how language and web spidering technologies can be applied to derive measures for them. Language technology is a wide area of research whose techniques can be used for analyzing the trustability of a text from a linguistic point of view. Web spidering is the programmatic analysis of the content and of the links between websites, used whenever structured information is required that is published on web sites and when that information is not available in a structured format.

Contact Information. The availability of contact information indicates openness and verifiability. Contact information can be available in the text itself, on the same web-page, or elsewhere on the site on which the text is published.

Grammar and Spelling. This indicator concerns the extent to which a text adheres to grammar and spelling rules. The rationale is that texts that adhere to grammar and spelling rules more closely are also more likely to have been written with greater care, and may therefore be more trustworthy.

Duplication. This indicator concerns the extent to which (parts of) the text occurs in other texts and locations. The rationale is that texts of which multiple copies exist have been copied, and must therefore be considered valuable and trustworthy. Automatic plagiarism detection is a research area within language detection that addresses this issue.

Publication Date. The publication date is the date on which the text was published. Related information that may be of interest is the date of the latest change. For some types of content, this may have implications for the trustability. Also, this information may be used to determine whether the information is out-of-date. E.g., information published a decade ago on topics that have seen a lot of changes in the last few years are less likely to be trustworthy. The publication date may be obtained directly from the text itself of by spidering the web page.

Source Reputation. This indicator concerns the reputation of the source: the author, the author's organization, and/or the publisher of the text. If the source is known and trusted, then the text may be trusted as well. In order for the name of the author, organization, or publisher to be determined, one may use named entity recognition. This is a technique that analyzes a text to extract proper names. A good example of source reputation tool is WikiScanner (wikiscanner.virgil.gr)

Subjectivity. Natural-language texts convey emotions and opinions about people, politicians, products, companies, *etc.*. A text's subjectivity may be used to determine whether one will trust the information. A text may have a negative subjectivity or a positive subjectivity on a given topic, and the level of subjectivity may vary from low to high. Sentiment analysis (a.k.a. opinion mining) [11] analyses texts for their emotional content, mostly to determine whether this is positive or negative attitude. For example 5 derives a "Dutch subjectivity lexicon" automatically. Instead, [14] presents a way to automatically determine subjective words and collocations (e.g., "unwise in", "ad hoc", and "drastic as") from corpora. Unique words we found to be subjective more often than expected. A good example of subjective analysis tool is given by the Dutch website Vox-Pop (vox-pop.nl/), which performs sentiment analysis on thousands of Dutch-language web pages with news and opinions on news.

2.3 Quality of Recommendation Analysis

Modern recommender systems must cope with an increasing demand of complexity; for instance, a recommendation application for restaurant should take into account the contextual information (e.g., has the restaurant been recommended for a romantic dinner or for a business lunch?). To provide a guideline for the evaluation of how recommendations provided by the recommender system can be trustworthy for the users, we have analyzed a large number of recommender sites and we have identified

a list of indicators (also cf. [1]) for the trustworthiness of their recommender or review services. We also considered the solutions that have been presented in relevant conferences (e.g., RecSys and IFIPTM conferences). The list is still incomplete and it will be extended as future work.

Robustness of Rating System: This criterion focuses on the strategies used to provide recommendations and ratings, and on the solutions that the web-site maintainers apply to avoid attacks that tries to subvert the fairness of reviews and consequently to bias recommendations.

Multi-criteria reviews. This criterion focuses on the number and on the type of the criteria used to rate the quality of a product or a service. More and more web site are now moving from offering only one general overall "quality" rating towards the use of multi-criteria. For example E-bay (www.ebay.com) has changed its rating system by asking buyers to enter four ratings corresponding to four different criteria about sellers (communication, shipping, speed & charges, and description adequacy).

Recognition and Roles. This criterion focuses on evaluating the availability of different roles in the users. For example, recommenders can be classified depending on their activities (e.g., sellers or buyers) or depending on their recognized reliability (e.g., experts or simple users) as in CNET.com (www.cnet.com). The criterion also describes if the recognition mechanism specifies how roles are managed, for instance, how a user changes role, if a certification is required to be addressed as an expert etc, as done by Dooyoo (www.dooyoo.com).

Rewards. This criterion evaluates the presence of rewarding mechanisms that stimulate recommender in leaving ratings and textual comments. An example is given by Epinions (www.epinions.com) that encourages members leaving good quality reviews, by rewarding them with Eroyalties, which are redeemable in U.S. dollars tracks how much a member earns for writing reviews. These bonuses are not tied directly to product purchases, but are based instead on more general use of reviews by consumers when they making decisions. Thus a member could potentially earn as much for helping someone make a buying decision with a positive review as he could for helping someone avoid a purchase with a negative review.

Personalization. This criterion evaluates the presence of mechanisms for the personalization of ratings, i.e., the presence/absence of solutions that help users understand whether a given opinion matches their taste, scope, context, etc. For example, users can block certain opinions because inappropriate, or unconvincing. Alternatively, users can perform searches based on their profiles or preferences. Explainability, that is the presence of a written explanation why a certain good has been recommended, is also a way to improve the degree of personalization [9].

Web of Trust. This criterion evaluates the presence of solutions that facilitate the establishment of a user's trusted network of recommenders, or alternatively the presence of solutions for the evaluation of the quality of the source of recommendations. Many review systems start offering their users an ad-hoc Web of Trust, composed by the network of reviewers whose reviews and ratings have been consistently found to be valuable by that member. The Web of Trust mimics the way people share word-of-mouth advices every day, and it is based on sociological concepts. The scientific community is devoting more and more attention to the role of trusted recommenders

in improving the accuracy of recommendations (cf. [8]). It has been proved that trust-based recommendations are both competent and well intentioned; the resulting recommender system is more accurate and robust [3].

3 Conclusion and Future Work

We started with an important question in mind: how can we to provide a scheme for the evaluation of the trustworthiness of a recommender site? In fact, not all the word-of-mouth is equal and consumers need to distinguish between 'good' (honest, true) and 'bad' (dishonest, false) information before deciding upon a purchase. Therefore the issue "*how consumers choose their information source and the mechanisms that help them find trusted information sources will be of particular interest for future research*" is of paramount importance for the success of a recommender site.

Recommender sites were born with the goal of helping users to cope with the information complexity typical of the Web 2.0 paradigm, but recommender systems' qualities and efficiencies also depend upon different factors. In this preliminary study we have identified and commented some of the most common indicators of trustworthiness. We addressed the following different "perspective" upon a recommender site: user interfaces, automatic content analysis, and quality of recommendations. We also found that techniques from language technology and web spidering can be used to support higher-level trustability analysis. Existing systems such as Melanie Martin's system and WikiScanner implement some of these ideas. The following table summarized our preliminary check list of relevant indicators for trustworthy recommender sites, according to the criteria we have introduced so far.

This checklist should be considered an early attempt to classify the variables or the factors that affect the perception of trust in the design of trustworthy e-commerce and recommender websites. Hence, validation is required to assess its applicability in different contexts and to test its comprehensiveness. Such a validation exercise would be the next step of our research. Nevertheless, knowing what factors determine trust does not suffice: We need to research ways to visualize these aspects in order to aid users in their assessment of the trust-related variables mentioned in Table 1. How can we present them? What effect does this have on purchasing behavior and on perceived

Table 1. Preliminary check list for trustworthiness evaluation of recommender sites

	User Interface	**Content Analysis**	**Recommendations**
Trustworthiness indicators	Certification from trusted parties; Management infos; Past reviews patterns; References from past and current users; Offer/Compare reviews; Source of reviews present and easy to check; Reputation of the reviews managed.	Contact info; Grammar/Spelling; Duplication; Publication date; Source reputation; Subjectivity.	Robustness; Multi-criteria reviews; Recognition and roles; Rewards mechanisms; Personalization; Web-of-Trust.

ease of use? And what role can social communities play with regard to trust? Finally, an interesting future work is to understand how to incorporate these variables into the design of recommender systems. Here, we suggest the need of tools for web design that support software engineers in automatically including "trust-enhancing" features in their products during the development phase.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Cassell, J., Bickmore, T.: External manifestations of trustworthiness in the interface. Comm. of the ACM 43(12), 50–56 (2000)
3. Dell'Amico, M., Capra, L.: SOFIA: Social Filtering for Robust Recommendations. In: IFIP Int. Federation for Information Processing, Trust Management II, vol. 263, pp. 135–150. Springer, Boston (2008)
4. Hertzum, M., Andersen, H.H.K., Andersen, V., Hansen, C.B.: Trust in information sources: seeking information from people, documents, and virtual agents. Interacting with Computers 14, 575–599 (2002)
5. Jijkoun, V., Hofmann, K.: Task-Based Evaluation Report: Building A Dutch Subjectivity Lexicon (D-14), Technical Report, Version 1, September 2 (2008)
6. Kelton, K., Fleischmann, K.R., Wallace, W.A.: Trust in Digital Information. J. of American Society for Information Science and Technology 59(3), 363–374 (2008)
7. Chopra, K., Wallace, W.A.: Trust in Electronic Environments. In: Proc. of the 36th Hawaii Int. Conf. of Systems Science (HICCS 2003), p. 331.1. IEEE Computer Society, Los Alamitos (2003)
8. Lathia, N., Hailes, S., Capra, L.: Trust-Based Collaborative Filtering. In: IFIP Int. Federation for Information Processing, Trust Management II, vol. 263, pp. 119–123. Springer, Boston (2008)
9. Koren, Y.: Tutorial on Recent Progress in Collaborative Filtering. In: Proc. the 2008 ACM Conf. on Recommender Systems (RecSys), Lausanne, Switzerland, October 23-25 (2008)
10. Nass, C.I., Fogg, B.J., Youngme, M.: Can Computer be teammates? Int. J. of Human-Computer Studies 45(6), 669–678 (1996)
11. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
12. Shneiderman, B.: Designing trust into online experiences. Comm. of the ACM 43(12), 57–59 (2004)
13. Veltmaat, L.: Taaltechnologie voor betrouwbaarheid van informatie, TR2009-01-12, Telematica Instituut (2009)
14. Wiebe, J., Wilson, T., Bruce, R., Bell, M., Melanie, M.: Learning Subjective Language. Computational Linguistics 30(3), 277–308 (2004)
15. Yang, J., Wang, J., Clements, M., Pouwelse, J.A., de Vries, A.P., Reinders, M.H.T.: An Epidemic-based P2P Recommender System. In: Proc. ACM SIGIR Workshop on Large Scale Distributed Systems for Information Retrieval (LSDS-IR 2007), Amsterdam, The Netherlands (2007)

Comment Classification for Internet Auction Platforms*

Tomasz Kaszuba[1], Albert Hupa[2], and Adam Wierzbicki[1]

[1] Polish-Japanese Institute of Information Technology Warsaw, Poland
kaszubat@pjwstk.edu.pl, adamw@pjwstk.edu.pl
[2] Institute of Applied Social Sciences, The University of Warsaw, Poland
albert.hupa@gmail.com

1 Introduction

Reducing Internet auction fraud is one of the greatest challenges in today's electronic market. Most of the electronic auction platforms use only simple reputation system that can be easily manipulated [1,2,3]. Although reputation systems can be used to detect frauds, they provide little detailed information about the fraud itself except user comments.

Despite many help pages and tutorials provided by auction platforms, in most cases it is not easy to teach users how to protect themselves from Internet fraud. To inform their users, auction services are offering insight into other user feedbacks. Yet, a large number of feedbacks presented to the user is sometimes an obstacle, rather than a support for the decision making user. Different users can have different opinions about the behavior of another user, but reputation systems treat every feedback equally. Thus it becomes necessary for the decision making user to read and analyze every comment, sometimes even proceeding recursively in order to evaluate how realiable the commenting user is.

We have developed a hierarchical model of user behavior in Internet auctions (separately for buyers and sellers) that will allow a meaningful distinction of different types of negative and neutral comments. The classification uses classes that have a clear interpretation for users, and that allow to evaluate the harmfulness of another user's behavior. The classes are created using both a Top-down and a Bottom-up method through an analysis of comment contents. The proposed classification method has been evaluated on a large trace from a real Internet auction site. We have also proposed method for rating complaints against sellers and buyers that can be used to modify the Internet auction reputation algorithms.

The rest of this paper is organized as follows: in the next section we describe related work. In section three, we discuss the characteristics of users in electronic auction market and their risk. In section four we propose the classification of complaints for seller and for buyer. Section five describes the classification results for real data from Internet auction platform. In section six we propose a system of rating of feedback types depending on the harmfulness of reported behavior. Section seven concludes and presents ideas for future work.

* The work reported in this paper has been funded by the Polish Ministry of Science and Higher Education under the research grants N N516 4307 33 and 69/N-SINGAPUR/2007/0.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 129–136, 2010.

2 Related Work

Most of recent work has been focused only on the seller's profile [4,5]. Much work has been devoted to inducing users to behave properly [6,5] as well as detecting fraudulent users [2,1]. There are some tools dedicated detecting fraudulent sellers (*ProtoTrust*[1]) or entire cliques of fraudulent agents (*NetProbe* [7]). Gavish and Tucci [3] have presented the seller's swindling methods in Internet auctions. Gregg and Scott [4] have proposed a model of complaints against sellers. Although their model is similar to ours, they have used a manual process to classify feedbacks and did not propose a rating of feedback types. The work of Dellarocas [6] applies in situations where users can intentionally give unfair ratings to each other. The author has proposed to conceal the identities of buyers and sellers to prevent such discrimination.

3 Characteristics of Agents in the Electronic Auction Market

We can distinguish three types of agents in Internet auctions: buyers, sellers and the auction service provider. Each type of agent has different interests and can execute different actions in the auction system.

- **The Buyer**
  The buyer is most vulnerable to fraud, because of the online auction architecture which in most cases requires the use of the advance payment method. Sometimes items can be paid by cash on delivery which is safer for the buyer. In general the buyer is obliged to make the payment before receiving the item. Hence a buyer's risk is much higher than a seller's.
- **The Seller**
  The sellers usually have a better position, because they do not risk any money, but the time spent on maintaining an auction indirectly affects their income. According to regulations sellers cannot interfere in their auctions, and they cannot refuse to sell the item if the auction is finished. In some cases a seller can revoke the bid of a user for a specific reason, but in most cases the seller has to deal with the winning buyer. If there is no payment after an appropriate time the seller can put this item up for auction once again. However, the seller has lost time on maintaining the auction as well as the handling fee. In some cases (specified by the auction platform) sellers can get their handling fee back.
- **The Service Provider**
  The third agent - the auction service provider risks no money, but its income depends directly on the total number of auctions carried out by sellers. Moreover there is a possibility (for example when the buyer does not pay for an item) that the seller can demand his handling fee back. Thus it is in the best interest of the auction service provider to discourage agents from cheating and punish frauds as quickly as possible.

[1] utrust.pjwstk.edu.pl

4 Feedback Classification Model

In order to create our classification model, we have obtained a real world dataset. The dataset has been acquired from *www.allegro.pl* which is the leading Polish online auction provider. We have selected the subset of 15159 negative or neutral feedbacks for 12188 different users. We have partitioned the feedbacks into two groups (for sellers and for buyers) and designed two independent classification rules for each group.

We have mined the information from the users' comments using two independent classification rules for each group - *top down* and *bottom up*. These approaches helped us to compare the outcomes - different types of complaints, on the basis of which we created a taxonomy by connecting the types according to different meanings.

4.1 Classification Methods

We have used two approaches to create the taxonomy of user complaints. In the first approach In the second approach we have used advanced data mining techniques to cluster the co-occurring words into groups. Then we have confronted the results from both methods and created the tree structures presented in Figure 1 and 2.

In the *top down* classification approach we have created a simple typology tree by a semi-automatic method using our *regex creator* tool. Tool has a mechanisms for creation of a regular expressions and assign new patterns to complaint types. The tool still needs human control to find a new pattern.

In the *bottom up* approach we have detected groups of words which frequently exist together. In order to do so we applied the Newman Girvan algorithm [8] [9] for community detection. This approach is based on the measures of shortest paths and betweenness centrality calculated for edges. The effect of the application of Newman Girvan algorithm consisted of sets of words which usually occured together in our dataset. These sets were treated as meaningfull types of complaints.

4.2 A Taxonomy of User Complaints

Complaints Against the Seller. The full model of complaints against sellers is presented in Figure 1. We distinguish two kinds of losses due to fraud: time and money related. We mark complaints related to loss of time with striped lines. Those colored in light-grey are related to loss of money. We have observed that there are two general groups of complaints: seller behavior related and item related.

- **Fraudulent behavior.** Shill bidding or shipping overcharge. We consider only explicitly formulated accusations, not those computed from historical auction data.
- **No response.** Communications with the seller after the auction was impossible. The seller did not answer phones, nor responded to e-mails.
- **Odd behavior.** The seller behaved in a completely unpredictable manner, communication with the seller was possible but handicapped. The seller sent the item with a delay or did not define the payment method and shipping price.
- **Item not sent or lost.** The item was not sent to the recipient. Sometimes the seller argues that the item was lost by the courier or post office.

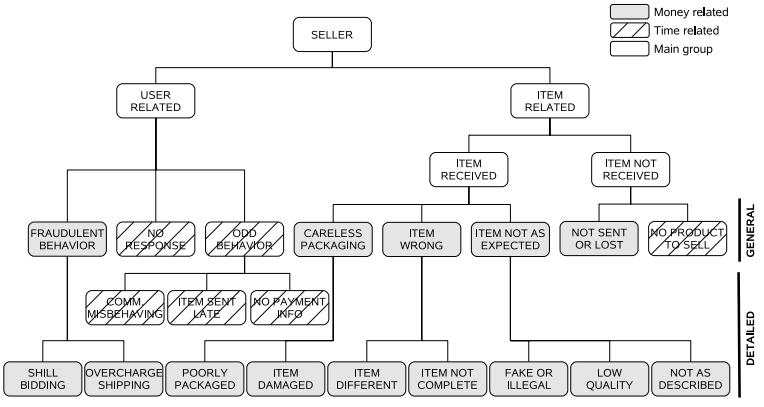


Fig. 1. Typology of complaints against seller

- **No product to sell.** The seller declares that the item was already sold to another buyer, or the item is no longer on sale. In this case the item is not sent to the buyer.
- **Careless Packing.** The seller did not take care about the packaging of the items. This type also includes the situation when the received item was damaged. It is not possible to verify if the seller sent a damaged item or the item was destroyed during shipment.
- **Wrong item.** The seller made a mistake and sent a wrong item (wrong color or type) or the received item was not complete.
- **Item not as expected** The item seems to be illegal goods (a fake, or a pirate copy of software) or just does not satisfy the buyer.

Complaints Against the Buyer. In Figure 2 we present the complaints model for the buyer. Similarly to the previous model we mark with striped lines complaints related to loss of time. Those colored in light-gray are related to loss of money. We can also partition complaints into user related and item related.

- **No response.** Communications with the buyer after the auction was impossible. The buyer did not answer phones, nor responded to e-mails. Complaints of this type are in most cases also classified as 'no payment' complaints (every complaint could be classified into more than one type).
- **Odd behavior.** The buyer seems not to follow the auction rules, or did not read the information provided by the seller. Sometimes the buyers tries to force the buyer to choose a particular payment method.
- **Delivery not accepted.** The buyer did not accept the delivery which should be paid for by cash on delivery. The seller must pay the round trip shipping charges, which is sometimes a significant amount of money. This is the only type of complaints against the buyer related to loss of money.

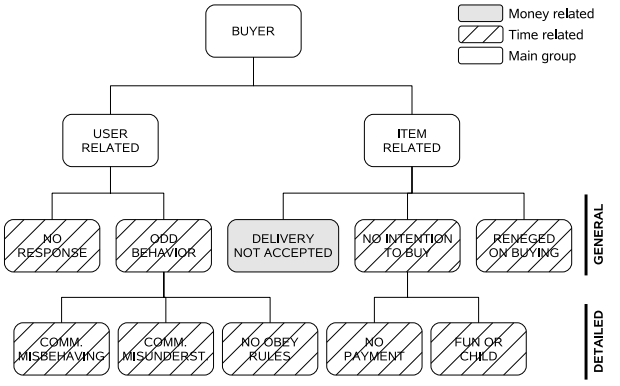


Fig. 2. Typology of complaints against Buyer

- **No intention to buy.** The buyer did not pay for the item, and did not inform seller about her plans. Sellers call such behavior childish or bidding for fun.
- **Reneged on buying.** The buyer contacts the seller and declares that she will not buy the item.

5 Classification Results

We have partitioned all negative and neutral feedbacks into the detailed types of the complaint taxonomy, using regular expressions prepared by the two classification methods. Each complaint type has its own meaning and also a unique set of regular expression patterns. In our evaluation we have used only types from the general level of the taxonomies, in order to obtain more legible results. Patterns from the detailed types are used in types from the general level. We have tested all negative and neutral feedbacks made by sellers and buyers and assigned to types in our taxonomy (for the seller and the buyer respectively). We have matched each feedback against all patterns from our model. A feedback could be assigned to more than one pattern from different types. We present normalized results of all neutral or negative feedbacks separately.

Our regular expression tool has matched 68% of negative comments (for the seller and the buyer equally), 54% of neutral comments for the seller and 35% of neutral comments for the buyer. Unclassified comments contain mostly useless information (no specified reason or lots of spelling errors). The amount of such feedbacks can be reduced by enabling users to choose one of our proposed complaint types from a list instead of editing comments by themselves, keeping the possibility of editing comments afterwards to add more information if desired.

The difference in classification quality between negative and neutral feedbacks is caused by the fact that neutral comments contain less complaints which are the most useful information for classification.

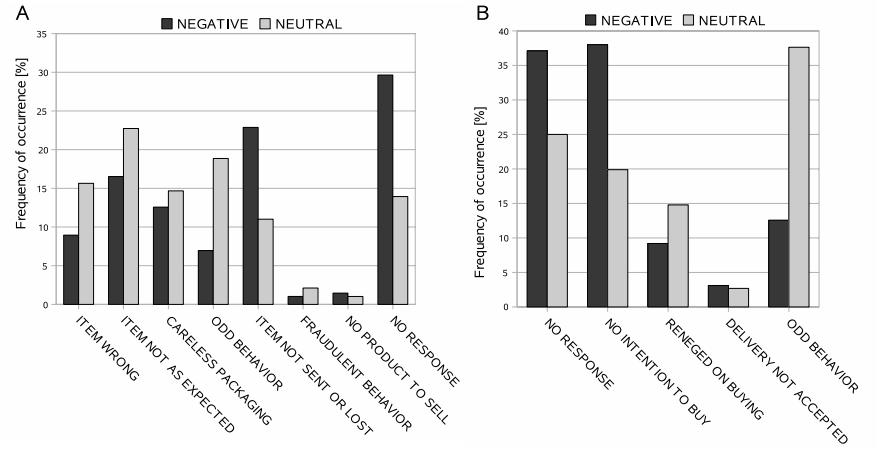


Fig. 3. a) Results for seller complaints, b) Results for buyer complaints

5.1 Classification of Complaints against the Seller

Negative feedback. In Figure $3a$ we present the frequency of occurrence of complaints against the seller. Most of the negatives are due to a lack of response from the seller or not receiving the item (Please compare it to the taxonomy presented in Figure 1). This is predictable since users do not like to be uninformed, especially when they risk their money. A significant amount of negative feedback is due to problems with the item, like sending a wrong or low quality item.

Neutral feedback. Neutral feedback was sent in most cases when the item did not live up to buyers expectation or the item was different (for example different color or size) than described in the auction. Seller behavior such as problems with understanding the seller or delays in sending the item was also a frequent reason for a neutral, rather than negative feedback. In comparison with negative feedback we can observe a significant drop (almost 50%) of complaints related to not sending the item or ignoring the buyer.

5.2 Classification of Complaints Against the Buyer

Negative feedback. We present the classification results for the buyer in Figure $3b$. Similar to the results for the seller, most of negative feedback was sent due to problems of communication with the buyer. There have been two main reasons to send a negative comment: the first is the lack of payment, the second is no communication at all (which often occurs simultaneously). We can observe a significant drop in the amount of negative feedback when the buyer declares that he will not buy the item (for any reason).

Neutral feedback. As we can see sellers tolerate all strange behavior of buyers as long as they pay for the item. They are also tolerant when the buyer declares explicitly that

he resigns from buying the item (item can be put for auction once again). In comparison with negative feedback we observe a considerable drop in the amount of neutral feedback when communication with buyer was not possible and thereby also no payment at all has been made.

6 Rating the Harmfulness of Unfair Behavior

To make our research more applicable to Internet auctions we propose a simple method for rating the types of complaints along their harmfulness. We propose *harmfulness* to be the difference between the frequency of occurrence of negative and neutral feedback. We compute the *harmfulness* for every type in our complaint taxonomy. A type of complaints tends to be more harmful if more negative than neutral feedback is classified into that type. We have sorted the groups of complaints along the *harmfulness* and present the detailed results in Table 1. We have also juxtaposed the *harmfulness* with the frequency of occurrence of each type of complaint. Values of the frequency of occurrence were generated from nonpositive feedbacks (negative or neutral feedback). In addition we have added the relation of each type to losses of time or money from the model presented in Figures 1 and 2. Our rating scheme does not need to be approved as is, but it can be used to detect major threats. We suggest that every user tunes this scheme to her preferences.

The most harmful seller behavior is lack of response (23%). To reduce this kind of unfair behavior, auction platforms can provide additional channels of communication with the seller. Another type of harmful behavior is not sending the item after the auction. This type can be reduced by charging the seller an amount which depends on the final price of an item, and to return this amount after the transaction completes successfully.

Table 1. Rating and the frequency of occurrence of types of nonpositive feedback

Complaint type against seller	Harmfulness [%]	Time or Money related	Frequency of occurrence [%]
NO RESPONSE	15.71	T	23.48
ITEM NOT SENT OR LOST	11.86	M	18.22
NO PRODUCT TO SELL	0.44	T	1.29
FRAUDULENT BEHAVIOR	-1.09	M	1.46
CARELESS PACKAGING	-2.1	M	13.4
ITEM NOT AS EXPECTED	-6.22	M	18.96
ITEM WRONG	-6.7	M	11.58
ODD BEHAVIOR	-11.9	T	11.62
Complaint type against buyer			
NO INTENTION TO BUY	18.11	T	36.47
NO RESPONSE	12.11	T	36.09
DELIVERY NOT ACCEPTED	0.42	M	3.07
RENEGED ON BUYING	-5.59	T	9.67
ODD BEHAVIOR	-25.06	T	14.7

The most harmful buyer behavior is bidding without intention to pay for the item and lack of response after the end of an auction. The joint frequency of occurrence of both types is 72% of all non positive feedback against the buyer. A good idea can be to introduce some time threshold after which the seller can automatically put the item for an auction again without paying the handling fee.

7 Conclusion and Future Work

In this work, we have presented a taxonomy of complaint types for buyers and sellers in Internet auctions. Our model is based on real data from *www.allegro.pl*. We have also proposed the rating of complaint types which can be a building block for an improved reputation system. Our rating scheme may be used by Internet auction platforms to detect and fight against the most harmful frauds and thereby gain more trust from the users. It can be deployed alternatively to the user's feedback list.

We are currently integrating our model with the *ProtoTrust* tool which is an interactive web browser extension that helps user in decision making process using trust management techniques. Through the integration with *ProtoTrust* we hope to create a helpful, user-friendly tool that can help users to detect unreliable contractors.

References

1. Rubin, S., Christodorescu, M., Ganapathy, V., Giffin, J.T., Kruger, L., Wang, H., Kidd, N.: An auctioning reputation system based on anomaly detection. In: CCS 2005: Proceedings of the 12th ACM conference on Computer and communications security, pp. 270–279. ACM, New York (2005)
2. Chau, D.H., Faloutsos, C.: Fraud detection in electronic auction. In: European Web Mining Forum at ECML/PKDD (2005)
3. Gavish, B., Tucci, C.L.: Reducing internet auction fraud. Commun. ACM 51(5), 89–97 (2008)
4. Gregg, D.G., Scott, J.E.: A typology of complaints about ebay sellers. Commun. ACM 51(4), 69–74 (2008)
5. Resnick, P., Zeckhauser, R.: Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In: Baye, M.R. (ed.) The Economics of the Internet and E-Commerce. Advances in Applied Microeconomics, vol. 11, pp. 127–157. Elsevier, Amsterdam (2002)
6. Dellarocas, C.: Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In: EC 2000: Proceedings of the 2nd ACM conference on Electronic commerce, pp. 150–157. ACM, New York (2000)
7. Pandit, S., Chau, D.H., Wang, S., Faloutsos, C.: Netprobe: a fast and scalable system for fraud detection in online auction networks. In: WWW 2007: Proceedings of the 16th international conference on World Wide Web, pp. 201–210. ACM, New York (2007)
8. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS 99(12), 7821–7826 (2002)
9. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69(026113) (2004)

ProtoTrust: An Environment for Improved Trust Management in Internet Auctions*

Tomasz Kaszuba, Piotr Turek, Adam Wierzbicki, and Radoslaw Nielek

Polish-Japanese Institute of Information Technology
Warsaw, Poland
kaszubat@pjwstk.edu.pl, piotr.tuek@pjwstk.edu.pl,
adamw@pjwstk.edu.pl, radek@post.pl

1 Introduction

Internet auctions are used everyday by millions. However, despite frequent criticism, only the most simple reputation systems are used by the most popular Internet auctions today. As a consequence of this, an experienced auction user is forced to undergo the menial task of reading and judging comments about his potential transaction partners. While it is true that the human mind is the best possible method of evaluating this information, the task is time-consuming and error-prone: the sheer number of comments is sometimes an obstacle to making a good decision under uncertainty. An inexperienced auction user, on the other hand, is often daunted by the task of understanding the information provided to him. It takes some learning to understand the value of a negative feedback, to evaluate the contents of the comments, or to understand what it means that the auction system has returned the handling fee. The reason for this situation may be the fact that the management of auction sites uses other mechanisms, like auction insurance or escrow, to protect users against outright fraud. And, as has been argued by the management of auctions sites when we have had an opportunity to discuss the issue, the simplicity of the presently used reputation system is an added bonus: it creates the impression of a simple, easy-to-understand tool. The fact that this simple tool is vulnerable to several adversary strategies [1,2,3] and that its design has an adverse impact on the reporting behavior of users [4,5] is not a sufficient argument for a change.

Our goal has been the improvement of trust management for Internet auction users. In our view, the trust management (TM) system should have as a goal the simplification of the users' search for relevant information, reducing the time complexity of the task of browsing through all relevant feedbackand increase the safety and comfort of the user. To solve this problem, we have designed an extension for a popular Web Browser (Firefox[1]) that gives users access to our algorithms. The algorithms themselves are part of a library of trust management tools developed in the uTrust[6] project. The extension obtains its information

* The work reported in this paper has been funded by the Polish Ministry of Science and Higher Education under the research grant N N516 4307 33.

[1] http://www.mozilla.org/

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 137–144, 2010.

by automatically performing the task that is performed by an auction user: by crawling parts of the auction site. The results of the crawling provide input information for the improved trust management algorithms. The user is presented with a graphical interface that gives access to a wealth of information that should support her in making the right decision.

We have decided to choose simple, yet useful algorithms for the first suite of tools provided by our extension. We have tested our algorithms on extensive traces of Internet auction use.

The rest of this paper is organized as follows: in the next section, we present the algorithms that have been implemented in the first version of the extension. Section three describes the evaluation of the proposed algorithms using traces of Internet auctions. Section four concludes and presents ideas for future work.

2 Related Work

In the area of Internet Fraud, most of recent work has been focused on the seller's profile [7,8]. Much work has been devoted to inducing users to behave properly [9,8] as well as detecting fraudulent users [2,1]. The work of Dellarocas [9] applies in situations where users can intentionally give unfair ratings to each other. The authors have proposed to conceal the identities of buyers and sellers to prevent such discrimination. Gavish and Tucci [3] have presented the seller's swindling methods in Internet auctions. Gregg and Scott [7] have proposed a model of complaints against sellers.

Currently, some stand-alone applications aim to solve a similar problem [2] [10], but as long as they have no integration with web browsers they are not user friendly. Since ProtoTrust uses the Firefox extension mechanism to install and update itself, it is easily accessible to all potential on-line auction users.

3 ProtoTrust Algorithms

We have designed several methods to help the users with decision to buy an item from a certain seller or not. The methods perform three tasks: they support the decision of the user based on various TM algorithms, they increase the amount of information available from the auction site by considering implicit non-positive feedbacks, and they perform a classification of auction comments to support the user in understanding them. In this paper we focus on the decision support algorithms that aim to directly recommend whether the user should buy an item from a specific seller. Two types of decision support methods are used by ProtoTrust: price dependent and probability dependent methods. Both types of methods take into consideration user preference such as *risk propensity* R_{prop} that is an amount of money that a user is willing to risk, or a *risk threshold* R_{thres} that is a threshold level for the probability methods. The exact meaning of these values depends on the used algorithm and will be explained further. Probability dependent methods are based on a proportion between count of negative feedback to all feedback collected by an agent. These methods are

similar to original reputation methods used currently in on-line auction services. In addition, we formulate decision rules for all the algorithms that may help the user to make decision about participating in an auction or discarding it.

3.1 Decision Support Methods

The main focus of this paper are decision support methods that use various TM algorithms to recommend the right decision to the user. In this section, we describe the TM algorithms implemented in uTrust for this purpose, and used in ProtoTrust.

Fraud Probability. Fraud probability $FraudProb$, or simply reputation, is the standard measure provided by any Internet auction service. It is a proportion between the number of negative feedbacks M and total number of feedbacks N.

We compute several variations of this measure that depend on the time class of the feedback. For seller s the fraud probability is defined as:

$$FraudProb_{st} = M_t/N_t \tag{1}$$

where M_t is amount of seller's s negative feedback not older than t, and N_t is total count of feedback not older than t.

ProtoTrust warns the user about this seller's auction when:

$$FraudProb_{st} > R_{thres} \tag{2}$$

This means that seller s has carried out too many fraudulent operation in time t. The value of R_{thres} is the threshold chosen by the user from the range of $[0, 1]$.

The fraud probability $FraudProb_c$ for every category c is computed in a similar manner, but we do not use it in this algorithm since we want to pick out every possible fraudulent seller. On the other hand, we use $FraudProb_c$ in other algorithms described below.

Reputation with Price Context. Many researchers [4,5,11,12] have proven that the reputation of a seller is related to selling prices. Therefore we propose our measures which can be complementary to the standard reputation.

We propose to compute the weighted average price for the sellers' auctions. $AvgPrice_s$ - is the seller's s weighted average price, in which weights are dependent on the value of the auction's feedback. For each seller's auction i we multiply the final price P_i and the buyer's feedback value $FVal_i$. Weights are -1 for negative, 0 for neutral and 1 for positive feedback. Let n be the number of auction carried out by seller s. Average price for the seller s is given by the equation:

$$AvgPrice_s = \frac{\sum_{i=0}^{n}(FVal_i * P_i)}{n} \tag{3}$$

We are not willing to compute the average price for the entire category too often, due to the computational cost. Since most of the auction systems are mature markets, the value of the average price for a category does not change frequently.

We compute the average price and standard deviation σ for each category using the full TM system information. These values may be computed once a fixed period of time and kept as constant values in the TM system.

When the user's $AvgPrice$ is significantly lower than the $AvgPrice$ in a context, there is a possibility that the seller will cheat in such an auction (by selling cheap items and not sending them to buyers or selling defective or illegal goods). Thus our system alerts the user by testing the seller s in a category c when:

$$AvgPrice_s + R_{prop} < AvgPrice_c \tag{4}$$

where R_{prop} is the *risk propensity* parameter defined by the user.

If we include the standard deviation σ we get:

$$AvgPrice_s + R_{prop} < AvgPrice_c + \sigma \tag{5}$$

However sometimes it is hard to point out a fraudulent seller basing only on his transaction history. Such sellers can establish a certain level of reputation before carrying out fraudulent auctions. Most of them gain reputation by selling many low-cost items. Note that one positive feedback from a €1000 auction is worth the same as from a €1 auction.

To protect from such cheating techniques we propose to compute the minimal price with a negative feedback $MinPriceWithNeg_s$. We should be wary of all offers from seller s which are much above the minimal price with some parameter R_{prop} which is the *risk propensity*. Our system alerts when the actual bid in an auction i is higher than the seller's s minimal price with a negative feedback. There is no reason to alarm when user has no negative feedback.

$$P_i - R_{prop} > MinPriceWithNeg_s \tag{6}$$

Risk. The measures proposed above may be not understandable for an inexperienced user. Sometimes it is more convincing for a user to compute the amount of money she can lose if the seller is fraudulent. Our risk measure $Risk_i$ is the multiplication of the actual bid P_i by the fraud probability in a context. It is given by the equation:

$$Risk_i = P_i * FraudProb_c \tag{7}$$

We compare risk to the risk propensity R_{prop} that is the amount of money that a user wants to risk in an auction. A user can set her risk propensity value to tune the TM system to her preferences.

$$Risk_i > R_{prop} \tag{8}$$

Our system alerts when $Risk_i$ is greater than a user's risk propensity R_{prop}.

4 ProtoTrust Algorithms Evaluation

We have evaluated ProtoTrust using a real world dataset. The dataset has been acquired from *www.allegro.pl* that is the leading Polish on-line auction provider.

In this service, each auction has an explicit deadline and all current bids are exposed to all participants. Moreover, all information about all participants is accessible.

We have selected the subset of 9500 sellers and their 186000 auctions listed in 6300 categories. We have tested our decision support algorithms using all 328000 feedbacks that are sent by the buyers. The unequal amount of auctions and feedbacks is caused by existence multi-item type auctions.

4.1 Experiments

We have reimplemented some of the uTrust algorithms to work with our off-line data. To recreate the on-line environment, we have sorted the auctions according to the termination date. For each auction in the set we have computed all the algorithms using only the data that was available until that moment. After computing all algorithms we have tested if they are good predictors of the real feedback value. For each algorithm we store: the count of successful detections of negative feedbacks $True_{neg}$ (the accuracy of the algorithm), and the count of unsuccessful detection $False_{neg}$ (Type II error in statistics) of negative feedback.

4.2 Evaluation Criteria

For the evaluation criteria of our algorithms we use two values: the probability of fraud detection FrD (recall) and frequency of alerts FoA (precision).

Let N be the total number of feedbacks and M the total number of negative feedbacks. Fraud detection is given the by equation:

$$FrD = True_{neg}/M \tag{9}$$

and the frequency of alerts is given by:

$$FoA = \frac{True_{neg} + False_{neg}}{N} \tag{10}$$

We have also computed the difference between fraud detection FrD and frequency of alerts FoA. We have used the random classifier as a reference level. For the random decision the FrD and FoA are equal (for example if we choose to alert in 50% of cases we discover 50% of true negatives $True_{neg}$).

4.3 Evaluation Results

We have evaluated the algorithms presented in the previous section from two different perspectives: probability dependent and price dependent. Results in each group depend on the user preferences (risk threshold R_{thres} and risk propensity R_{prop}). For better presentation, we have selected three best algorithms from each group. We have presented detailed results achieved by all algorithms in Table 1.

Table 1. Best performance achieved by algorithms

$Algorithm_type$	R_{prop} $[PLN]$	R_{thres} [‰]	FrD	FoA	$Performance$ $(FrD$ - $FoA)$
$FraudProb_inf$	–	5	0.42	0.12	0.3
$FraudProb_2$	–	5	0.35	0.09	0.26
$FraudProb_4$	–	5	0.41	0.12	0.29
$Risk$	1	–	0.48	0.07	0.41
$AvgPrice$	19	–	0.45	0.18	0.27
$MinPriceWithNeg$	19	–	0.51	0.23	0.28

Probability Dependent Methods. For probability dependent algorithms, we have run the experiment several times, changing the risk threshold parameter R_{thres}. R_{thres} is the acceptable probability of fraud and it is expressed in permils [‰]. On Figure 1*a* we present the effect of the risk threshold R_{thres} on detection of fraud FrD and frequency of an alert FoA.

Best fraud detection was achieved by the algorithm that used all available feedbacks (*FraudProb inf*). However, this algorithm also had a high frequency of alerts. Moreover, in a real situation we would not gather all historical data about the seller because of time and network usage. Similar results have been achieved using only feedbacks that are not older than 4 weeks (*FraudProb 4*). Using feedbacks that are at most 2 weeks old (*FraudProb 2*) gives us a 10% lower detection rate, but also has a much lower frequency of alerts. Both Fraud Detection and Frequency of Alerts decline linearly with increasing of risk threshold R_{thres}. When we increase the risk threshold, our system is less likely to alert the user about fraudulent sellers, because it accepts some sellers' negative feedbacks.

As shown in Table 1, the best trade-off between fraud detection and frequency of alerts was achieved when the risk threshold value was fixed at 5 ‰. This is because we observe a significant drop of the frequency of alerts and slight decrease of fraud detection when this value of the risk threshold is exceeded.

Price Dependent Methods. Similarly to the previous methods, we have run the experiment several times, with different risk propensity R_{prop} parameter values.

We have selected three algorithms described in 3.3S. Figure 1*b* presents the fraud detection FrD and frequency of a alerts with regard to the risk propensity R_{prop}. Best results have been achieved by $Risk$. This algorithm has detected every fraudulent auction while it was alerting every second auction. With an increase of the risk propensity, the algorithm detects less fraudulent auctions. We can observe a drastic gap between fraud detection and frequency of alerts for $Risk$ when R_{prop} equals to 1 PLN (1 Polish Zloty is about €0,25). The algorithm can eliminate half of possible fraudulent auctions while alerting only in 7% of all offers. The two other algorithms provide similar a detection rate with a much higher frequency of alerts (about 25%). We have modified the $AvgPrice$ algorithm (described in 3.3) by including information about sellers' minimal

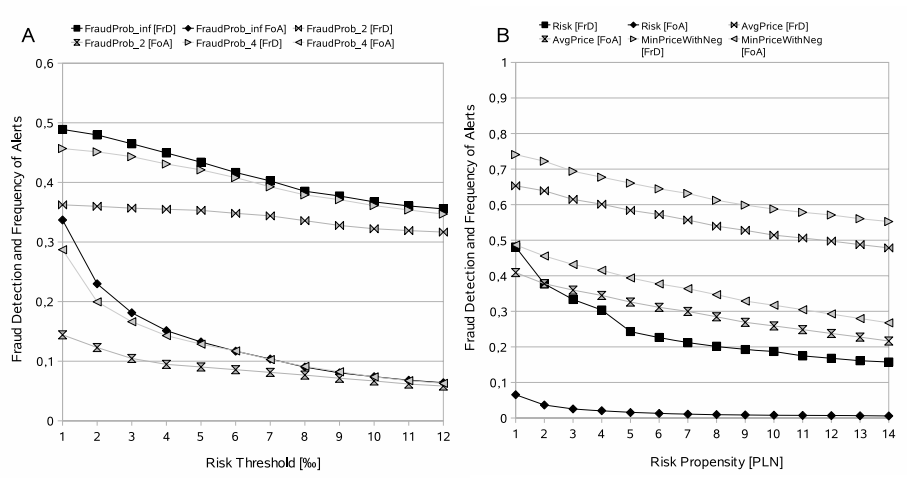


Fig. 1. Probability of Detection of Fraud FrD and Frequency of an Alert FoA in respect of: a) risk threshold R_{thres} [‰] and b) risk propensity R_{prop} [PLN] (Polish zloty)

price with a negative feedback. As a result, the $MinPriceWithNeg$ algorithm has a slightly better score than the original $AvgPrice$.

The best trade-off between FrD and FoA was achieved by $Risk$ with a risk propensity equal to 1. It is 41% better than the random algorithm. Detailed results are presented in Table 1.

The risk algorithm tends to be the most effective. It provides a good fraud detection rate with a very low frequency of alerts. When the risk propensity is between 1 and 4PLN (€1), the risk algorithm performs better than any other presented algorithm. Using this measure we can warn user against almost half (48%) of the frauds before the they occur.

5 Conclusion and Future Work

In this work, we have presented ProtoTrust, an environment for improving Trust Management for Internet auctions that operates independently of the auction providers. We have designed new algorithms that detect Internet auction frauds and proposed the decision making rules that help users to detect fraudulent sellers. We have evaluated our algorithms on real data from the largest Polish Internet auction provider (Allegro), and have shown that we can protect users from almost half of fraudulent auctions when ProtoTrust alerts users in only 7% of auctions.

In the future we are planning to adapt our ProtoTrust environment to work with other major auction services. We plan to distribute the crawling tasks between all active instances of ProtoTrust to increase the efficiency.

References

1. Rubin, S., Christodorescu, M., Ganapathy, V., Giffin, J.T., Kruger, L., Wang, H., Kidd, N.: An auctioning reputation system based on anomaly detection. In: CCS 2005: Proceedings of the 12th ACM conference on Computer and communications security, pp. 270–279. ACM, New York (2005)
2. Chau, D.H., Faloutsos, C.: Fraud detection in electronic auction. In: European Web Mining Forum at ECML/PKDD (2005)
3. Gavish, B., Tucci, C.L.: Reducing internet auction fraud. Commun. ACM 51(5), 89–97 (2008)
4. Melnik, M.I., Alm, J.: Does a seller's ecommerce reputation matter? evidence from ebay auctions. Journal of Industrial Economics 50(3), 337–349 (2002)
5. Ye, Q., Li, Y., Kiang, M., Wu, W.: The impact of seller reputation on the performance of online sales: evidence from taobao buy-it-now (bin) data. SIGMIS Database 40(1), 12–19 (2009)
6. Kaszuba, T., Rzadca, K., Wierzbicki, A., Wierzowiecki, G.: A blueprint for universal trust management services. Technical report, Polish-Japanese Institute of Information Technology, Warsaw, Poland (2008)
7. Gregg, D.G., Scott, J.E.: A typology of complaints about ebay sellers. Commun. ACM 51(4), 69–74 (2008)
8. Resnick, P., Zeckhauser, R.: Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In: Baye, M.R. (ed.) The Economics of the Internet and E-Commerce. Advances in Applied Microeconomics, vol. 11, pp. 127–157. Elsevier Science, Amsterdam (2002)
9. Dellarocas, C.: Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In: EC 2000: Proceedings of the 2nd ACM conference on Electronic commerce, pp. 150–157. ACM, New York (2000)
10. Pandit, S., Chau, D.H., Wang, S., Faloutsos, C.: Netprobe: a fast and scalable system for fraud detection in online auction networks. In: WWW 2007: Proceedings of the 16th international conference on World Wide Web, pp. 201–210. ACM, New York (2007)
11. Grtler, O., Grund, C.: The effect of reputation on selling prices in auctions. Discussion Papers 114, SFB/TR 15 Governance and the Efficiency of Economic Systems, Free University of Berlin, Humboldt University of Berlin, University of Bonn, University of Mannheim, University of Munich (May 2006)
12. Schlgel, C., Wolff, B.: Country-specific effects of reputation and information: A comparison of online auctions in Germany, the UK, and the US. FEMM Working Papers 07027, Otto-von-Guericke University Magdeburg, Faculty of Economics and Management (December 2007)

Extending Trust in Peer-to-Peer Networks

Stephen Clarke, Bruce Christianson, and Hannan Xiao

School of Computer Science, University of Hertfordshire, UK
{s.w.1.clarke,b.christianson,h.xiao}@herts.ac.uk

Abstract. This paper presents a way of reducing the risk involved with downloading corrupt content from unknown (and hence untrusted) principals in a P2P network. This paper gives a brief overview of the need for trust in P2P networks, introduces a new notion called trust*, and shows how this may be used in place of the conventional notion of trust. Finally, we apply trust* to the Turtle P2P client and show how the integrity of downloaded content can be guaranteed without assuming that trust is transitive.

1 Introduction

Peer-to-peer (P2P) based networks are widely used on the Internet to enable file sharing, streamed media and other services. With a traditional client-server based network, many clients connect to a fixed server. Whereas P2P clients are all considered equal and connect directly to each other. Because of this topology, tasks such as sharing files and other resources can be more efficient as a client can connect to many other clients and download content simultaneously.

Much of the content currently distributed via P2P networks is either illegal or violates copyright laws in some way. However, there are also many legitimate reasons why content might be distributed using P2P, and there is copyright-free content also available such as open source software. P2P protocols such as BitTorrent enable sharing of very large files such as operating systems, and many Linux based distributions are downloadable in this way in order to lower the load on an individual server.

P2P networks have many advantages such as scalability, and due to there being no centralised server, network loads can be easily balanced. However, for the same reasons, a problem with P2P networks is that all peers are regarded as equal and there is no real way to moderate content. Anyone can use a P2P client and share any files they wish. Malicious users can easily insert incorrect files into a network which are searchable by other clients and will therefore propagate further. Even non-malicious users might be unaware that they are serving incorrect files from their computer. To counter this, hosts might publish an MD5 check-sum on their website. However, this is unlikely and it is the user's decision whether and how they actually verify this, and getting hold of the correct checksum leads us back to the initial problem. Also, this approach assumes that the trustee is the original source and not just a middleman provider.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 145–152, 2010.

This paper describes how a new concept called trust* [3] can be applied to P2P networks to guarantee the integrity of files being shared. This paper uses the Turtle P2P client [11] as a basis on which to discuss the concept, although trust* can be applied to various other P2P clients. Turtle enables files to be shared among friends (people whom you know in the real world) in the hope to improve safety and overall integrity of the shared content. However, friendship isn't transitive. Trust* aims to reduce the perceived risk involved when sharing files over multiple hops with unknown principals. Trust* achieves this by providing incentives to act correctly and deterrents for acting maliciously or incompetently.

2 Extending Trust

This section briefly describes the concept of trust* [3]. The main purpose of trust* is to allow unknown principals to interact whilst at the same time lowering the perceived risk incurred by transitively trusting or relying on reputation (particularly when the intention is to use a client *once* and never again).

In the real world, this is often achieved by using an intermediary as a guarantor. An example of this is letting houses to students, where landlords require a guarantee against a particular tenant. The guarantor trusts the tenant and the landlord trusts the guarantor so the landlord has shifted the risk of not receiving the rent to the guarantor. The landlord believes that he will always get his rent whether it be from the tenant or the guarantor.

Trust* is based on the electronic equivalent of the real world guarantee solution. Say that Alice needs to trust Carol about something and doesn't personally know or trust Carol. However, Alice trusts Bob who in turn trusts Carol to do whatever it is Alice needs her to do. In order to change Alice's perception of the risks involved, Bob could guarantee to Alice that Carol will act as intended and offer Alice compensation if Carol doesn't. The concept of "extending" trust in this way by using localised guarantees is what we call a trust* relationship.

The trust*er (Alice) can then act as if they trust the trust*ee (Carol) directly. In order to shift the risk, forfeit payments are used. All forfeits are paid locally; if Carol defaults then Bob must pay Alice the agreed forfeit whether or not Carol pays Bob the forfeit she owes him (and the two forfeits may be of different amounts). Failure to provide a service – or to pay a forfeit – may result in an update to a *local* trust relationship; for example, between Bob and Alice, or between Carol and Bob. Figure 1 illustrates a typical trust* relationship.

Trust* can be composed to an arbitrary number of hops because all trust is now local and so are the forfeits. It is worth noting that trust isn't the same as trust* even in a one hop scenario; in this case, if Bob trust*s Carol to provide a service, it means that Bob trusts Carol to either provide the service or else pay the forfeit[1].

[1] Bob may believe that Carol cannot provide the service, but will always pay the forfeit.

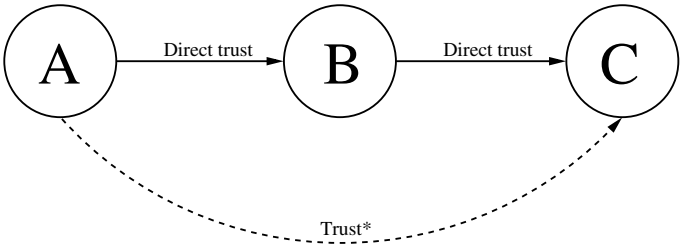


Fig. 1. A trust* relationship

3 Trust in P2P Networks

Due to the nature of P2P networks and the likelihood that interactions are between completely unknown and untrusted principals, peers in a network need a way to mitigate the risks they would incur if they temporarily trust others. The risks involved are likely to vary depending on what is actually being shared. For example, software should be the correct version and should not be corrupted in any way, documents should be authentic and music should be licensed.

There are many security and trust issues related to P2P networks [1,5,9,13] and the trustworthiness of others is normally gauged using some kind of reputation system [8,12]. However, reputation systems have a vital flaw; they imply that trust is always transitive [6] which can be a bad assumption [2]. Assume a user wants to determine the risk involved if they were to trust another (eg. to provide a described service) by looking at their reputation rating. This might contain comments and ratings left from previous transactions. It is unlikely that the user looking knows (or trusts) the other users who have left the comments. Also, reputation systems are prone to threats such as Sybil attacks [4] where the same user can operate under many pseudonyms. But even if a user does know and trust the people who left the comments, they will still be transitively trusting the service provider in question.

According to Jøsang *et al* [7], transitivity is possible with the correct combination of the referral and functional variants of trust. However, trust* allows the risk involved to be underwritten, even when these delicate conditions for transitivity are not satisfied. With trust*, Bob is not only making a recommendation to Alice, but also offering compensation if something goes wrong. The trust scope is decided locally between Alice and Bob when the guarantee is created. It is assumed that the final guarantor in a trust* chain will have functional trust in the end-point (or trust*ee).

Most services provided over a distributed system or network have (as in the real world) an underlying contract or agreement. In most cases, this could simply be that service X will be provided for a fee P and that the service will conform to the terms and conditions of X. In P2P networks, such guidelines do not at present generally exist and clients connect to other clients to become an equal part of the network. Peers are usually free to download anything they wish from

other peers and vice versa. There may be situations where content could be charged for or where a particular service level agreement is in place, however, it is more likely that peers in a P2P network hold a "download at your own risk" policy regarding the files that they are sharing.

Trust* can be deployed to provide the missing assurance when indirectly trusting others. For example, Carol doesn't care if someone wants to download file X and doesn't care if they are unhappy with it. However, Bob has previously downloaded files from Carol, and hence trusts that her files are of a high standard. Alice trusts Bob so Bob's guarantee reduces the risk for Alice. If Bob was wrong, he will compensate Alice with the agreed forfeit. However, in this example, Carol hasn't necessarily done anything wrong and isn't obliged to reimburse Bob. Bob however is likely to lower his high perception of the quality of Carol's files and perhaps never guarantee her again. Bob's motivation to provide the guarantee is a commission payment from Alice[2]. Bob will set the level of this commission depending on his perception of the probability of Carol defaulting[3].

4 Applying Trust* to Turtle

The Turtle client requires you to list your friends whom you trust to share files with. The Turtle protocol works by only sending queries for files to these friends, who pass on the query to their friends as their own query and so on[4]. Such queries and their results are only ever swapped within these local trust relationships. The second stage is for the original requester to choose the file to be downloaded from the list of results. The file is then downloaded locally by each peer in the chain in the same manner as the search query.

This localised trust setting is perfect for also finding routes of trust* guarantees, as the query and result route used could also make up a chain of guarantees. Extending the example to a longer chain, Alice wants to download file X and sends a query to Bob whom she trusts. Bob forwards this query to Carol whom he trusts. Carol continues to forward this to her friends. Dave receives the query, he has file X and sends back a positive response to Carol which is forwarded back to Bob and then Alice. Assuming now Alice chooses Dave's file via Bob from the list of search results and requests that it comes with a guarantee from Bob, a guarantee chain could be negotiated at the same time as retrieving the file. The scope of the trust* guarantee is also negotiated between each pair which states the terms of the guarantee and what constitutes a breach.

Suppose Alice discovers that the file X is corrupt in some way. Alice can claim the forfeit from Bob. Bob may also claim from Carol. Suppose Dave does not

[2] In a commercial case, where Carol provides a service for payment, Carol may pay Bob a commission for acting as an intermediary.

[3] Provided Bob's estimate of the probability of Carol defaulting is lower than Alice's *a priori* estimate, then both Alice and Bob will be happy with the guarantee.

[4] If you have read the spam-proof application in [3], please note that the direction of trust in that case goes in the opposite direction to that described here for Turtle. Trust* works perfectly in either direction.

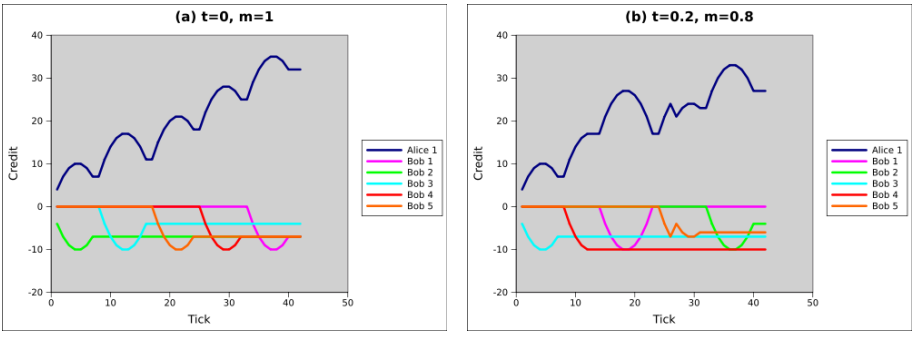


Fig. 2. Principals exhibiting bad behaviour

care if his files are correct. So rather than Carol claiming from Dave, she is likely to stop trusting him altogether, or not guarantee against him again, or charge a higher commission from Bob in future for providing the guarantee.

Eventually, say that Dave is habitually sharing corrupt content, all principals who once trusted him are likely to never guarantee his files again. In a fair P2P system where credit or reputation is gained depending on the quantity of uploaded content, and is used to download files from others, Dave will also have trouble buying guarantees from others (or they will be very expensive for him). In this example, the commission can be thought of as an insurance payment.

Alternatively, someone might guarantee only certain types of files from another peer. For example, Carol might be happy to guarantee any of Dave's music files but considers the software that he shares as risky so Carol will not guarantee these files. Trust* can enable these fine-grained decisions to be made. Even when Carol trusts Dave directly, she can still be selective over what she'll actually guarantee.

4.1 Simulation of Trust*

In order to analyse the effectiveness of applying trust* to a P2P scenario, the model was simulated with the Repast Simphony modelling toolkit [10]. A scenario where Alice wishes to download a file from Carol was simulated. There are five possible trust* routes (via the guarantors numbered 1 to 5) and each principal holds many properties including a credit rating[5]. Two global attributes t and m define the probability of Alice being truthful and Carol sharing incorrect files respectively. The trust* protocol is invoked once every "tick" of the simulation and stops when all available routes have been exhausted. The graphs in figures 2 and 3 show the resulting credit ratings for each principal and how long each simulation ran for.

In graphs (a) and (b), where Carol has a high chance of sharing corrupt files, the simulations stop after 42 ticks. By graphs (c) and (d), the probability of files

[5] This is purely to gauge the total gains and losses of a principal.

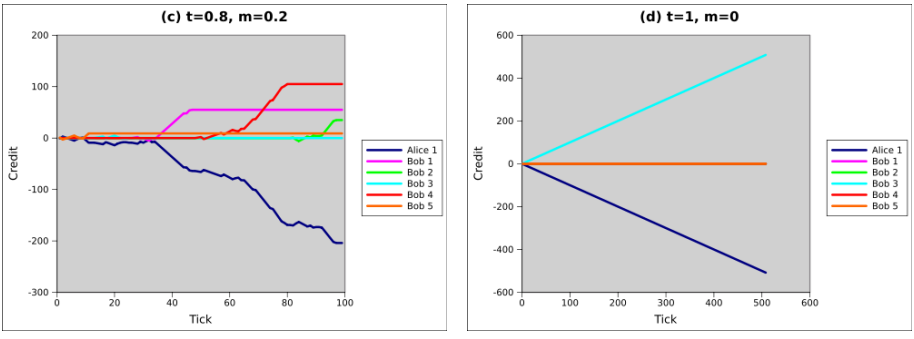


Fig. 3. Principals exhibiting good behaviour

being corrupt decreases, and hence, the simulation runs for longer. By graph (d) where the corruption chance is 0%, only one guarantor is ever used and the simulation would run forever. Many other graphs show fluctuations in forfeit rates and claims etc, however results presented here are limited for space reasons. The results show that long term trust* usage implies good behaviour from all involved principals. The guarantors will only tolerate misbehaviour for so long before refusing to provide further guarantees of the offending principal.

5 Discussion

5.1 Heterogeneity and Anonymity

In order to implement the trust* relationship mechanism, whether to initiate, provide, or receive a guarantee, a way of making decisions and payments is necessary. This functionality could easily be incorporated within P2P client software. One of the advantages of our approach is that the trust management and payment systems can both be heterogeneous, due to the fact that trust (and payments) are confined or localised. If a guarantee has been made from one principal to another, any trust management and payment schemes could be used between them. At the same time, other pairs of principals might use completely different schemes.

Because of this localisation of trust, end-point anonymity can also be maintained as principals only speak to their direct neighbours. No knowledge need be gained about other principals or the schemes they might be using. Also, participants need not know whom they are downloading from.

5.2 Payment by Resource

The most obvious use of a forfeit is either to deter a principal from defaulting on what they have guaranteed or to provide a way of compensating the other party if they do[6]. The commission payment was introduced in order to provide

[6] Note that these are slightly different requirements; a lower forfeit will often suffice for the first.

an incentive for a principal to act as a guarantor and can be seen as a spot price for a guarantee. A principal needing to trust* could pay this commission to a guarantor whom they trust directly. Forfeit and commission payments serve different purposes and don't need to be of the same type (or paid by the same means), although in the case of P2P networks, they could easily be.

Due to the heterogeneous nature of the localised trust between individual pairs of principals, the payments could take the form of a more immediately valuable commodity to them than a conventional micro-payment. In P2P file sharing applications, this could be the content itself. For example, credit to download further files or to buy licenses or guarantees.

6 Conclusion

This paper has presented the concept of trust* as a mechanism for guaranteeing the integrity of content or services provided over a P2P network. Trust* builds on the idea of sharing with friends in the Turtle P2P client but also guarantees the integrity of downloaded content from unknown peers derived through transitivity.

Using trust* in this way also reduces the risk involved for the downloader as they will be compensated in the worst case scenario. It therefore lowers the risk of transitively trusting others, and privacy is still maintained. This is because the guarantees and payments are confined within the same localised trust relationships as the ones that are used to communicate the actual search queries and their corresponding results. This approach therefore allows complete localisation of trust management, and the risk of trusting by referral is underwritten by the guarantees. We regard local trust management as a significantly easier problem than global reputation management, particularly in a P2P system where the majority of participants wish to be anonymous (except to their friends). As mentioned earlier, the use of trust* does not constrain the way in which local trust is managed.

Simulation of the trust* protocol shows that misbehaving principals quickly become isolated before major damage can be made. This means that threats such as a Sybil attack can be identified and the perpetrator will eventually be removed from local trust relationships. This will make it harder for them to share files in a P2P community that employs the trust* model as eventually all routes will be removed (or become too expensive).

The Turtle client was developed with an emphasis on privacy and safety of sharing files that might be of a controversial or provocative nature. Due to the localised direct trust in a trust* chain, such privacy can be easily maintained[7]. We have argued that applying trust* to P2P file sharing will also be beneficial in guaranteeing the integrity of free content such as open source software or copyright-free movies.

[7] However, privacy is not so much of an issue when sharing open content, and in other applications where the integrity of the content is more important.

References

1. Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-Peer Information System. In: Proceedings of the Tenth International Conference on Information and Knowledge Management, pp. 310–317 (2001)
2. Christianson, B., Harbison, W.S.: Why Isn't Trust Transitive? In: Proceedings of the International Workshop on Security Protocols, pp. 171–176. Springer, Heidelberg (1997)
3. Clarke, S., Christianson, B., Xiao, H.: Trust*: Using Local Guarantees to Extend the Reach of Trust. In: Proceedings of the Seventeenth International Workshop on Security Protocols (April 2009) (to appear)
4. Douceur, J.R.: The Sybil Attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, p. 251. Springer, Heidelberg (2002)
5. Jiang, J., Bai, H., Wang, W.: Trust and Cooperation in Peer-to-Peer Systems. In: Li, M., Sun, X.-H., Deng, Q.-n., Ni, J. (eds.) GCC 2003. LNCS, vol. 3032, pp. 371–378. Springer, Heidelberg (2004)
6. Jøsang, A., Gray, E., Kinateder, M.: Analysing Topologies of Transitive Trust. In: Proceedings of the Workshop of Formal Aspects of Security and Trust, pp. 9–22 (2003)
7. Jøsang, A., Hayward, R., Pope, S.S.: Trust Network Analysis with Subjective Logic. In: ACSC 2006: Proceedings of the 29th Australasian Computer Science Conference. Australian Computer Society, Inc. (2006)
8. Koutrouli, E., Tsalgatidou, A.: Reputation-Based Trust Systems for P2P Applications: Design Issues and Comparison Framework. In: Fischer-Hübner, S., Furnell, S., Lambrinoudakis, C. (eds.) TrustBus 2006. LNCS, vol. 4083, pp. 152–161. Springer, Heidelberg (2006)
9. Mondal, A., Kitsuregawa, M.: Privacy, Security and Trust in P2P environments: A Perspective. In: Bressan, S., Küng, J., Wagner, R. (eds.) DEXA 2006. LNCS, vol. 4080, pp. 682–686. Springer, Heidelberg (2006)
10. North, M.J., Howe, T.R., Collier, N.T., Vos, J.R.: The Repast Simphony Development Environment. In: Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms (2005)
11. Popescu, B.C., Crispo, B., Tanenbaum, A.S.: Safe and Private Data Sharing with Turtle: Friends Team-up and Beat the System. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols 2004. LNCS, vol. 3957, pp. 213–220. Springer, Heidelberg (2006)
12. Selçuk, A.A., Uzun, E., Pariente, M.R.: A Reputation-Based Trust Management System for P2P Networks. In: CCGRID, pp. 251–258. IEEE Computer Society, Los Alamitos (2004)
13. Wallach, D.S.: A Survey of Peer-to-Peer Security Issues. In: Okada, M., Pierce, B.C., Scedrov, A., Tokuda, H., Yonezawa, A. (eds.) ISSS 2002. LNCS, vol. 2609, pp. 42–57. Springer, Heidelberg (2003)

Business Rule Model Integration into the Model of Transformation Driven Software Development*

Olegas Vasilecas and Aidas Smaizys

Klaipeda University,
Herkus Mantas str. 84, LT - 92294 Klaipeda, Lithuania
Olegas.Vasilecas@ik.ku.lt, Aidas.Smaizys@ik.ku.lt

Abstract. Modern business is going to be global and more complicated, rapidly evolving and requiring frequent changes. This leads the software systems growing and going more complex, widely distributed and being pressed by business for continual changes at the same time. This result the need of new methods and methodologies used for modern software development. One of such relatively new additions in the field of traditional software development is business rules and model transformation driven approach. The paper discusses existing business information systems engineering issues arising out of such additions based on running researches in the field also carried out by the authors.

Keywords: business rules, model driven architecture, model transformation, software development.

1 Traditional Approaches of Software Development

The main success factor of software development is the selection of proper and efficient software development approach and the selection of suitable software development methodology. Such methodology defines software development process, methods being used and artefacts delivered in every step of the software development process. It provides a set of formal guidelines and instructions which define how all the software development process should be organized and executed.

Traditional software development process is based on iterative waterfall model of software development life cycle. This model contains iterative phases such as: conceptualization and requirement engineering, analysis and functional decomposition, design, coding, testing and deployment. All the traditional software development life cycle phases in some form could be found in both classes of software development approaches like heavyweight and lightweight (or agile) methodologies [1].

Nowadays most common heavyweight methodologies used in practice are based on Rational Unified Process (RUP) [2], Microsoft Solutions Framework (MSF) [3] and

* The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project "Business Rules Solutions for Information Systems Development (VeTIS)" Reg. No. B-07042.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 153–160, 2010.

lightweight (agile) methodologies are based on eXtreme programming (XP) [4, 5] or on some combination of them. Both the heavyweight and lightweight methodologies have limited flexibility and poorly adapt to the changing environment.

OMG initiative [6], The Model Driven Architecture (MDA) has shifted the focus of software development from writing code and documentation to building models and using them for generation of automated code or at least design specifications. However when organization considers applying MDA-based software development process it still faces the lack of methodologies implementing model-oriented software development paradigm.

The paper discuses development of agile software systems (SS) of business information systems (BIS) based on Business rule (BR) model involvement, transformation and integration into the model of traditional software system development to combine traditional software engineering processes, model-oriented software development paradigm and business rules approach and provide new methodology and methods suitable for intelligent software system development, ensuring business needs for rapid changes.

2 Business Rule Model Based Software Development Process

In the year 1992 Zachman with J. Sowa [7] extended the Zachman framework adding tree additional columns people, time and motivation. This was the first time when BR were explicitly introduced in the motivation column and became its place in process of IS engineering. The sixth - motivation - column of the framework represents BR aspect in business system and IS of the enterprise according to the different views: scope, business model, system model, technology model, components and describing the final implementation of BR in a newly functioning enterprise after system is implemented.

To describe BR model based software development process first of all we need to discuss definition of what we call functional and process views. From our opinion the main difference here is in the analysis perspective (Fig. 1). If we look at the system as a black box, then from outside of the system function view reflexes system outputs; and resource view reflexes external interfaces. This way process view will be the view into the inside of the box to analyse how resources from the input are processed into the results (services or products) on the system output.

Also it is possible to look at the inputs and outputs of the system from inside of the system. This can lead to different descriptions of the same things. For example we can look at the information system (IS) from different perspectives as in fig. 2 providing different models according to the Zachman Framework [7].

Or we can look at the IS as a part of BS, or as a set of SS's providing it's functionality as resources needed for IS (fig. 3). Different views and perspectives are the main issue of misunderstanding and confusing definitions of what BIS, IS and SS or plainly software really is.

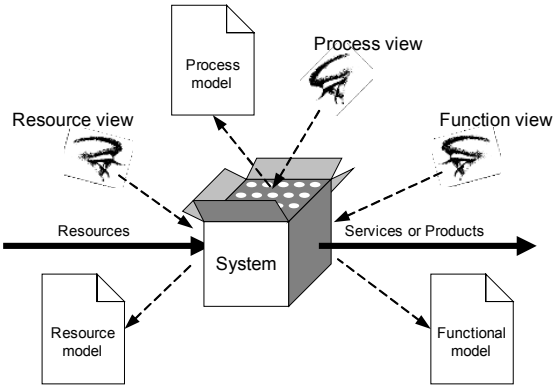


Fig. 1. System analysis perspectives - resource, process and function views

BIS is a system threaded through all the levels BS, IS and SS. First of all it is a part of BS providing functionality needed for business processes servicing all BS participants and evolving to automate more and more complex business processes growing from operational to strategic levels.

From IS view BIS is an IS providing data analysis and communication functionality for informational BS processes represented and modelled in IS level. Same time looking from the bottom, from developers perspective BIS consists of several software applications elaborated in SS, which provide resources needed for servicing informational processes of IS and BS.

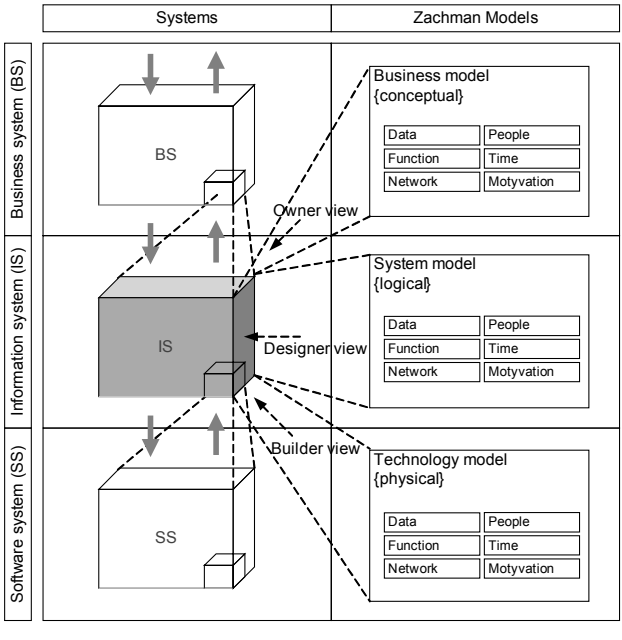


Fig. 2. Owner, designer and builder views to provide IS models

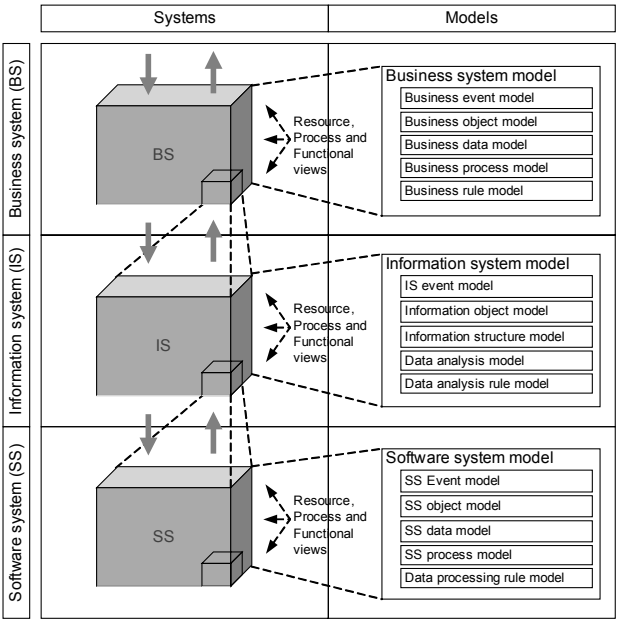


Fig. 3. Resource, process and functional views to provide BS, IS and SS models

Fig. 3 represents framework for BR model based BIS development process and models used. This framework is presented in detail in [8]. The main idea is in using metamodel based models and model transformations like in MDA-based software development process. However we use Business system (BS), Information system (IS) and Software system (SS) models as displayed in fig. 3 instead of Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM). We propose to develop several related models to reflect all the needed views and perspectives in every horizontal set of models or so called level of the models such as BS, IS and SS. Where Business rule model is a part of Business model and it is related to the Business object model which provides vocabulary and terms needed for specification of Business rules represented and stored in Business model. There are two main concepts of the proposed model framework – model relation and model transformation in both horizontal and vertical directions.

The same way as the transformations produced we can look at the separate systems such as business system, information system and software system to analyse inputs and outputs of each of them trading SS as a part of IS and IS as a part of BS. Where all of them are integrated and represented as one BIS, which represents not only software produced, but also shows how such software is integrated into the whole business system and how it is servicing informational business processes according to the business logic captured and represented in the separate BR model of Business system model.

From the software engineering perspective it is important to highlight that packaged BIS applications already have some embedded business rules technology. Rules in such packages usually cover basic business rules, but in case you have, for

instance, a complex financial or legal scenario you would need to extend rules and enable flexible BR enforcement. This could be allowed in Business rule based software systems, leading to the need of selection of corresponding architecture, components, techniques and tools. On the base of our experience we could define the following different strategies and BR approaches based on IS development maturity and used engineering process complexity:

- **Separation** of business logic model in early design process and software engineering process-driven BR implementation in the software system, typically using model transformations into the active DBMS component code dynamically or manually in the stage of system design;
- **Centralisation** of business logic execution in dedicated BR execution component using interpretation mechanisms of rule engine and BR enforcement in the integrated software systems activating built-in executable business processes for selected decisions;
- **Intellectualisation** of BIS by implementation of intelligent algorithms to allow system self adaptation involving automated business rule mining out from enterprise data and artificial intelligence;

Further in this section we will discuss all of these approaches of BR based BIS development and analyze their weak and strong sides according to the particular purposes.

2.1 Separate Development of Business Logic Model

Judging from the works of [9] we can draw a conclusion that the business logic exchange is vital for easy and fast software modification and adaptation to the business changes. There are several ways of separate development of business rule base business logic model development and implementation into the final software system proposed: to implement into relational database constraints [10], triggers [11, 12] or executable code in applications using resolution of separately stored BR and facts representing entered value instead of validation code [13, 14]. Usually this approach could be used for development of one particular component of the system, and achieved by separate development of static business logic model used in BIS for execution of structured decisions using software system parameterisation [15] and later reuse of components developed [16] and model driven service composition [17]. This approach is similar and usually does not need the use of rule engine. However inference processing and involvement of some rule engine is useful for validation of BR combined into the rule sets used for further automated transformations. Summarising our experience we could define two different alternatives of such BR enforcement and separately developed business logic model implementation:

- **Parameter driven approach** - system parameterization using BR and business logics usually represented in decision tables [18];
- **Model transformation driven approach** - Business rule and business process model automated transformation into the executable business processes and business logics exchange between separate parts of the business processes implemented into the several software applications later on [8];

Another alternative strategy of the business logic exchange is achieved by centralization of business logic execution presented in the next section.

2.2 Centralized Business Logic Execution

Business logic used in business processes is distributed through different information systems software and usually differs depending on technology used for each particular software system. This leads to the problems dealing with inadequate automated decision processing and implementation of changes into different software or several parts of the whole system. This can be avoided by using centralized business logic execution and BR enforcement in the integrated software systems according to the framework presented in [9].

The main advantage of this approach is that here the part of software dedicated to the business decisions together with the rules describing business logic is separated and implemented as the separate central component also called business rule management system (BRMS). This component is responsible for business decisions and includes a model of decision propagation into the components of enterprise software through several available information system interfaces. The central part of proposed framework includes some reasoning processor for business rule execution. For example, rule execution engine also called BRE carries out various actions based on the reasoning results in the internal inference engine, ensuring data transferred between any two integrated subsystems to conform according to the centrally stored BR and data entered by users representing facts.

Due to a flexible XML language used for BR formalization, it allows introducing an open organisation structure with a new intelligent functionality. Such a BR based meta-knowledge base system creates an open organization structure and allows a new intelligent functionality to enter the meta-knowledge and use it in all subsystems with corresponding rule exchange and enforcement interfaces to engage more duties in decision making process.

2.3 Intellectualisation of Information Systems

Summarising intelligent features of information systems that we can get from the approaches described in previous sections – they usually are acting by structured rules and do not deal with every possible impact on the business environment or future consequences. That is the main reason why automated decisions based on such rules can not take responsibility and requires involvement or approval of dedicated business people. This limits decision automation possibilities. The use of fuzzy logic instead or together with crisp logic in information systems can be used to simulate a real business environment and evaluate possible impact providing automated heuristic decisions.

During our research we have challenged a few problems not discussed before and related to the situations, when BR are not known at the beginning or they are incomplete even contradicting but we still need decisions. Moreover such BR system are not static and should be changed when a business situation changes. This means the need of development of the software systems that could be self adaptable to such changes and ensure information system adaptation to maximise achievement of goals

dedicated by the executive staff. Such situations may be resolved by involvement of fuzzy logic or flow engines implemented in BRMS component called BR mining engine. We have made some experiments for solution of such problem by building a risk model [14]. Similarly in [19] author uses statistical methods for simulation of decision influence on future state of business system. Such approach allows dynamical adaptation of the user interface of the software system application to the changes in business system observed from the collected enterprise data and generation of the business service and results in the extensions or modifications to the workflow logic, business logic or data logic being directly reflected in the operative user front-end and the presentation logic of information system.

3 Conclusions and Future Work

In this paper we have overviewed BR model and model transformation driven approach integration and separated three main strategies based on maturity and complexity of engineering processes and methods used. Depending on the approach it may be possible to ensure different level of agility by an instant deployment of changes in the Business policy and immediate reaction to the changes on the market or competition by changing existing business rules and introducing new rules not by programmers, but by business analysts. Such advances allow modern systems to be more transparent, auditable and to achieve cost reduction because of more efficient process of introduction of changes in business policy into the software systems of BIS used for implementation of decision automation and decision support.

BR based software systems have more complex development process in an initial phases, but such a system is more efficient in further maintenance and simplified modifications that is especially needed for businesses with frequently changing regulations and business policy, competitive behaviour on the market and requiring a high level of customisation and adaptation to the large scale of separate customer needs, especially when the software is designed using SOA.

References

1. Nikiforova, O., Nikulsins, V., Sukovskis, U.: Integration of MDA Framework into the Model of Traditional Software Development. In: Frontiers in Artificial Intelligence and Applications, Databases and Information Systems V, vol. 187, pp. 229–239. IOS Press, Amsterdam (2009)
2. IBM: Rational Unified Process: Best practices for software development teams (2003), http://www.ibm.com/developerworks/rational/library/253.html
3. Microsoft: Microsoft Solution Framework: MSF Process Model v. 3.1. White paper (2002)
4. Beck, K.: eXtreme Programming eXplained. Addison-Wesley, Reading (2000)
5. Martin, R.C.: RUP/XP Guidelines: Pair Programming. Rational Software Corporation (2002), http://rup.hops-fp6.org/papers/pdf/xppair.pdf
6. OMG: Model Driven Architecture, http://www.omg.org/mda
7. Zachman, J., Sowa, J.: Extending and formalizing the framework for information systems architecture. IBM Systems Journal 31(3), 590 (1992)

8. Vasilecas, O., Smaizys, A.: The Framework for Business Rule Based Software Modeling: An Approach for Data Analysis Models Integration. In: Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV, vol. 155, pp. 175–188. IOS Press, Amsterdam (2007)
9. Avdejenkov, V., Vasilecas, O., Smaizys, A.: Business Rule Management in Enterprise Resource Planning Systems. In: Frontiers in Artificial Intelligence and Applications, Databases and Information Systems V, vol. 16, pp. 255–266. IOS Press, Amsterdam (2009)
10. Zimbrao, G., Miranda, R., Souza, J.M., Estolano, M.H., Neto, F.P.: Enforcement of business rules in relational databases using constraints. In: Proceedings of XVIII Simposio Brasileiro de Bancos de Dados / SBBD 2003, pp. 129–141. UFAM (2003)
11. Valatkaite, I., Vasilecas, O.: Deriving active database triggers from business rules model with conceptual graphs. Lithuanian Mathematical collection 42, 289–293 (2002)
12. Sosunovas, S., Vasilecas, O.: Transformation of business rules models in information systems development process. In: Databases and Information Systems Doctoral Consortium: Sixth International Baltic Conference Baltic DB&IS 2004, Riga, Latvia, June 6-9, vol. 672, pp. 373–384. University of Latvia (2004)
13. Tang, Z., Maclennan, J., Kim, P.P.: Building data mining solutions with OLE DB for BM and XML for analysis, Rec. SIGMOD 34(2), 80–85 (2005)
14. Vasilecas, O., Smaizys, A.: Business Rule Enforcement and Conflict Resolution Using Risk Analysis. In: Proceedings of the 14th International Conference on Information and Software Technologies (IT 2008). Research Communications, pp. 92–98. Kaunas University of Technology, Kaunas (2008)
15. Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M.C.: Adaptive and Dynamic Service Composition in eFlow. HPL-2000-39. HP Lab. Techn. Report (2000), http://www.hpl.hp.com/techreports/2000/HPL-2000-39.html
16. Kilov, H., Simmonds, I.: Business patterns: reusable abstract constructs for business specification. In: Humphreys, P., et al. (eds.) Implementing Systems for Supporting Management Decisions: Concepts, methods and experiences, pp. 225–248. Chapman and Hall, Boca Raton (1996)
17. Orriens, B., Yang, J., Papazoglou, M.P.: Model driven service composition. In: Orlowska, M.E., Weerawarana, S., Papazoglou, M.P., Yang, J. (eds.) ICSOC 2003. LNCS, vol. 2910, pp. 75–90. Springer, Heidelberg (2003)
18. Smaizys, A., Vasilecas, O.: Agile Software System Development and Customisation using Business Rules. In: Frontiers in Artificial Intelligence and Applications, Databases and Information Systems V, vol. 187, pp. 243–254. IOS Press, Amsterdam (2009)
19. Graeme, S., Seddon, P.B., Willcocks, L.: Second-wave Enterprise Resource Planning Systems. Cambridge University Press, Cambridge (2003)

From Requirements to Code in a Model Driven Way

Audris Kalnins, Elina Kalnina, Edgars Celms, and Agris Sostaks

University of Latvia, IMCS, Raina bulvaris 29, LV-1459 Riga, Latvia
{Audris.Kalnins,Elina.Kalnina,
Edgars.Celms,Agris.Sostaks}@lumii.lv

Abstract. Though there is a lot of support for model driven development the support for complete model driven path from requirements to code is limited. The approach proposed in this paper offers such a path which is fully supported by model transformations. The starting point is semiformal requirements containing behaviour description in a controlled natural language. A chain of models is proposed including analysis, platform independent and platform specific models. A particular architecture style is chosen by means of selecting a set of appropriate design patterns for these models. It is shown how to define informally and then implement in model transformation language MOLA the required transformations. By executing these transformations a prototype of the system is obtained.

Keywords: model driven development, transformations, requirements, UML.

1 Introduction

The main goal of this paper is to demonstrate how transformations could be used to support the full path from requirements to code in a model driven development. Requirements are specified in the requirements specification language RSL [1,2] which has been developed as a part of the ReDSeeDS project [3]. A significant part of RSL is the specification of requirements to system behaviour in a controlled natural language. In this paper it is demonstrated how such requirements can be used as the basis for transformations to code via Analysis, Platform Independent (PIM) and Platform Specific models (PSM). Models are generated according to a particular architecture style, including the selection of appropriate design patterns for these models. Each transition in this chain is to a great degree assisted by formal model transformations. Though one specific chain of models is described here the approach could be applied to any similar setting of models.

In the Model Driven Architecture (MDA) approach [4] it is assumed, that the proper contents for CIM are requirements. Requirements model is built in a special requirement specification language (RSL) (see section 3). The required behaviour specification is sufficiently precise, therefore this specification can be processed by model transformations in order to generate initial versions of the next models.

The next models are built using subsets (and profiles) of UML 2[5]. In our extended MDA approach the next model is Analysis model borrowed from the standard OOAD methodology [6] (see section 5). The most important model in the proposed model

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 161–168, 2010.

chain is the PIM model (as in MDA). This model is built according to the selected design patterns (see section 2) and contains the description of structure and detailed behaviour of the system-to-be in a platform independent way. Transformations which generate the initial version of this model use both Requirements and Analysis as inputs. This is the step in the whole chain of transformations which contributes most to the rich system functionality inferred directly from requirements. The principles by means of which a nontrivial system behaviour description in PIM can be extracted from requirements in a controlled natural language with keywords are also the most innovative aspect of the paper. The contents of PIM are described in section 6.

The next model is the Platform Specific Model (PSM) fairly in standard MDA style (section 7). It is built by transformations from PIM by adding the platform relevant details. The paper demonstrates the combination of Java and Spring /Hibernate frameworks in the role of target platform. Finally, PSM is transformed to code (annotated Java EE in this case).

The main practical value of the approach is that a nontrivial executable prototype of the system can be obtained from requirements without manual extension of intermediate models. Certainly, a true model driven development should follow, where at each step the required details of the real system are filled in manually.

All model-to-model transformations in our approach are implemented in model transformation language MOLA [7] which occurs to be very appropriate for the given kind of tasks. The transformation development is discussed in section 8.

2 Principles of Model Structure Creation

Nowadays enterprise systems are developed using a set of design patterns. There are two types of design patterns: platform independent and platform specific ones. The traditional GoF design patterns [8] represent the former type. On the other hand, low level patterns such as the adequate usage of Spring framework are platform specific.

We use the concept of *architecture style*, which includes the system and model structure and the related set of design patterns. In this paper only one *architecture style* is discussed. The selection of architecture style is out of scope of the paper.

We have chosen four layer architecture, with the following layers: Data access layer, Business layer, Application logic and User interface. We have domain objects as data containers (“POJO”). Another general principle is that our approach is based on a declarative object-relational mapping (ORM). Persistent domain objects are treated as the basis for ORM definition. Whenever possible, we use the interface based design style. The design relies on the dependency injection pattern for referencing other classes. At the Data access layer Data access objects (DAO) are used for explicit ORM - related actions. DAO classes have CRUD operations and the standard transaction support. At the Business layer for each domain object participating in business logic a class is created, which encapsulates all business level operations related to this concept.

The application logic and User interface layers are governed by the MVC pattern. For application logic in addition the façade pattern [8] is used. Application logic operations are invoked by MVC controllers within this use case. UI part is kept as simple as possible. It contains only calls to application layer. All the above mentioned style elements apply to the PIM level.

Platform Specific design patterns are used in the PSM and in code. The POJO pattern is used in a most complete way, adapted to the Spring style. We use the declarative ORM definition based on annotations (coded as stereotypes in the PSM). The selected design patterns allow creating really usable code, not only code skeletons.

3 Requirements Model

The Requirements Specification Language (RSL) [1,2] is a semiformal language for specifying requirements to a software system. It employs use cases for defining precise requirements to the system behaviour. Each use case is detailed by one or more scenarios, which in turn consist of special controlled natural language sentences. The main type of sentences is the SVO(O) sentence [2], which consists of subject, verb and direct object (optionally, also indirect object). In addition to SVO(O), there can be also conditions, rejoin sentences ("gotos" to a point in the same or another scenario) and invoke sentences (invoke another use case). Alternatively, the set of scenarios for a use case can be visualized as a profile of UML activity diagram. RSL example is demonstrated in Fig.1. (For this to be a correct requirements model the relevant notions must also be defined.)

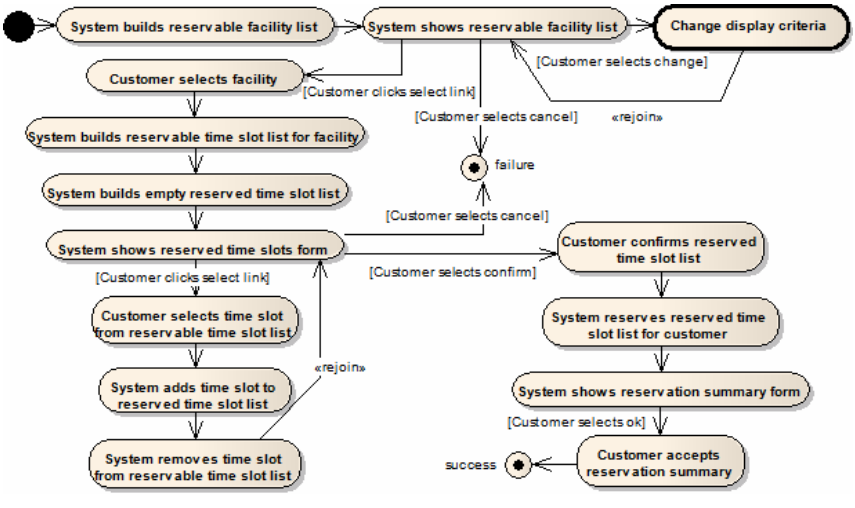


Fig. 1. RSL example from Fitness Club system, reservations - how a customer can book regular access to the selected fitness facility

Domain definition in RSL consists of actors, system elements and notions. Notions correspond to elements (classes) of the conceptual model of the future system. It is also possible to define generalizations and simple associations between notions.

The precise syntax of RSL is defined by means of a metamodel [1]. The behaviour and domain parts in valid RSL requirements model must be related. The subject of an SVO(O) sentence must be an actor or system element. An object must be a notion.

For the existing version of RSL a tool support [9] has been built. Currently RSL is extended by introducing keywords, which assign a predefined meaning to some verbs and nouns. This paper is based on the extended RSL version.

Transformations described in this paper can be applied to any valid set of requirements in RSL for a system. To generate something more than skeleton from such requirements some knowledge about meaning of theses sentences is required. To solve this problem transformations use keyword based heuristics for generating platform independent model. The chosen verb keywords for SVO(O) sentences are *show*, *select*, *build*, *add* and *remove*. The noun keywords are *form* and *list* – for use as parts of complex notion names. Conditions can contain the verb keyword *click* and *select*. The adjective *empty* is also treated as a keyword. The heuristics and choice of keywords depend on the selected architecture style.

4 Analysis Model

The Analysis model is generated by transformation from the domain (notions) part of Requirements. It creates classes from notions in Requirements. In addition to pure domain model of the system all design related concepts such as forms, their elements and ways of user interaction with the system are extracted from requirements and stored in the Analysis model. Certainly, all this is done in a platform independent way. Stereotypes are used to mark properties of classes, for example, <<*entity*>> (persistent class), <<*form*>>, <<*list*>>, etc. Some associations having a special meaning are also given special stereotypes (<<*Owned*>>, <<*ListItem*>> etc.).

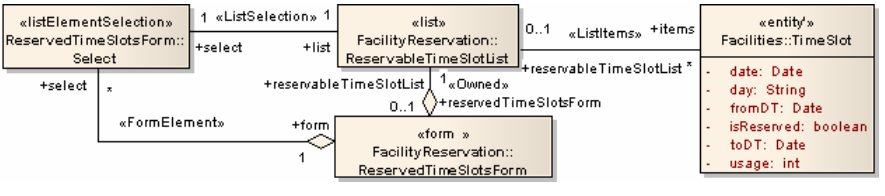


Fig. 2. Fragment of the generated Analysis model from requirements in Fig. 1

Class stereotypes are determined by analyzing keywords. However, some elements can be generated only by analyzing scenarios. For example, *select*-sentences (such as ... *selects* time slot from reservable time slot *list*) permit to conclude that the relevant form (that in the preceding *show*-sentence) permits to select elements exactly from this kind of list. Hence, this list (here, ReservableTimeSlotList) is visualized in the form (the <<*owned*>> association can be built), and the relevant selection element corresponds to an element of this list (the <<*ListSelection*>> association is built).

Using these and some other principles the Analysis model for the example (see Fig. 2) can be generated from notions and the scenario in Fig.1. This model can be extended manually in the Analysis step.

5 Platform Independent Model

This model is the most important one in our approach since the whole platform independent functionality is generated into this model. This is done by repeatedly analyzing use case scenarios taking into account the (possibly manually extended) Analysis model. In combination with the keyword based sentence analysis a significant part of application and business logic can be generated.

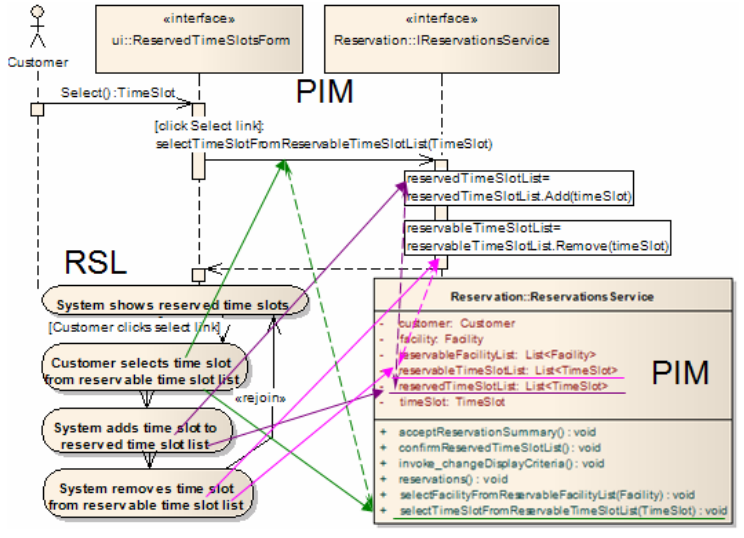


Fig. 3. Informal mapping example for transformations to PIM

One result of this step is the design class model: packages, interfaces and classes with all attributes and operations (with parameters), all annotated using stereotypes.

The class model is created according to the platform independent design patterns described in chapter 2. For each application logic class methods are generated on the basis of UI related scenario analysis. For business logic classes methods are created in the general context of behaviour generation, by analyzing scenarios in requirements. For each DAO class CRUD operations are needed. Bodies of these operations are similar for all classes, only types vary. Therefore we propose to implement them once in a template class, which contains parameterized types.

Another result is sequence diagrams, covering also significant part of business logic method bodies. All method invocations with appropriate parameters which can be generated are coded this way. Whenever possible, invocation logic up to DAO level is documented. In order to build an application logic method body, we look for consecutive scenario sentences with the subject System and recipient system (in other words, any verb other than "System *shows* …). All these sentences correspond to calls to the Business logic layer. The immediate recipient of this call depends on the sentence structure. For example, if the indirect object (e.g., …*for* facility) is present, the call is directed to the manager for the corresponding entity (here, FacilityService).

There are also other "patterns" for sentences which correspond to business logic calls The grouping of the generated business logic calls is done in a simple way – all these calls up to the next UI call (corresponding to the next "System *shows* …" sentence) are included in the body of the current application logic method body (see Fig. 3). The "System *shows* …" sentence generates a call to the user interface layer, which completes the current body.

Fig. 3 illustrates one typical application of the transformation rules described above by an informal "model mapping diagram", with arrows going from source model instances (bottom) to the corresponding target model instances (top). The second sentence in the scenario fragment ("Customer selects time slot from reservable time slot list") is an actor-to-system action, therefore it implies the method invocation *selectTimeSlotFromReservableTimeSlotList*() to the application logic class (*ReservationsService*). The next two sentences correspond to the actions in the body of this application logic method. These are list operations which are treated as elementary actions within the method body. *Add* and *remove* in this context are treated as keywords. The sentence "System shows …" is a system-to-actor sentence. There are some more rules in the approach quite similar to those explained on the example.

6 Platform Specific Model and Code

This model is a specialisation of PIM to a specific platform. Currently Java with Spring + Hibernate 3 is chosen, with declarative (annotation based) style. The main task is to create annotations according to the style required by Spring and Hibernate. However some new model elements should be added as well. For example, database diagram (with tables, columns, PK, FK etc) is generated from the domain objects.

Domain objects are used to describe Hibernate specific ORM functionality. All Hibernate and Spring specific annotations are added (coded as stereotypes) to domain classes, attributes and operations. Traceability links between PIM and PSM elements are generated by transformations and used to maintain various annotations related to mappings between different parts of the model.

For each DAO class the annotation <<@*Repository*>> and annotations describing the transactional mode are added. Application logic layer classes are included in the Business logic layer. Classes in these layers are given the annotation <<@*Service*>> (to mark them as Spring beans). The annotation <<@*Autowired*>> (Spring specific dependency injection) is used to initialize references to other beans.

For UI currently a rudimentary solution based on Spring MVC directly is incorporated. We use JSP for data visualisation and controllers to manage user actions. We use one controller per form, with a method for each user action. Typically a controller method directly calls the appropriate application logic method.

The provided PSM can be used for Java code generation. The structure of Java code will directly correspond to the structure of PSM. Method bodies form sequence diagrams are also generated. An initial version of configuration files, data base script and Java project for Eclipse IDE are also generated. All this provides a complete prototype – simple but ready to execute. In order to switch to other platforms, only the transformations from PIM to PSM and from PSM to Java have to be modified.

7 Implementation of Transformations

All described transformations have been implemented in transformation language MOLA [7, 10]. A transformation fragment can be seen in Fig 4. It represents a procedure for creating the UML lifeline corresponding to the call target.

The metamodel used for transformations (RSL + UML) is extended by special traceability elements. Transformations in every step build also the relevant traceability links. In addition to enabling the general traceability support in the toolset, traceability links are reused by transformations for finding the context of an element.

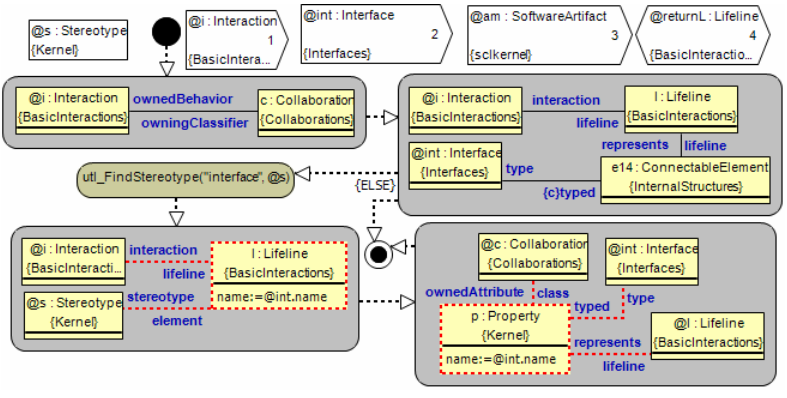


Fig. 4. Fragments of transformations in MOLA (to PIM)

All transformations in the chain must support repeated runs; therefore, support for various result merge actions is included. It mainly relies on traceability links.

The complete implementation of the described rules and merge facilities has confirmed the applicability of MOLA for the support of sophisticated model driven development. The most distinguishing quality of MOLA here appeared to be its easy readability which permits to adapt easily the transformations to changing models and informal rules.

8 Related Work and Conclusions

The CIM model [4] means requirements for the system, understandable by domain experts. Typically requirements are written as a free text, but a strong intention is to apply transformations also to CIM. Therefore a natural approach is to specify requirements in a controlled natural language, as is done in this paper. A similar kind of requirements are used as a starting point in [11,12,13]. The approach closest to ours is [11], where the Natural MDA language is proposed for behaviour description. This language uses a large set of keywords therefore it is much closer to programming languages than RSL, and the transformation based approach there is only partial. The approach in [12] is based on the Language Extended Lexicon and does not use the behaviour description thoroughly. An interesting approach is proposed in [13] where the initial requirements in natural language are manually converted into a list of semiformal functional features which then can be transformed formally using the topological model.

There is much work on transforming PIM to PSM, but this is not the main topic of our paper. To sum up, none of the mentioned approaches support a full path from textual requirements directly provided by domain experts to a system prototype where all transitions are performed by model transformations.

The paper shows the feasibility of a transformation supported path from semiformal requirements to code in a model driven way. The key aspects which have enabled this are the selection of an appropriate architecture style (the general structure and appropriate set of design patterns) for the system and an appropriate requirements language. Then a corresponding set of transformations can be defined which can extract maximum facts from requirements and convert them into appropriate elements of models in the development chain. The most crucial of models in the chain is the PIM. For building of this model the most sophisticated analysis of requirements has been done. The next model – PSM is adapted to the selected platform – Java, Spring and Hibernate. For models in the chain – Analysis, PIM and PSM appropriately defined UML profiles are used. The models obtained during this approach serve as the basis for further manual model driven development, using the same transformations for support. All the transformations are implemented in the model transformation language MOLA.

Acknowledgments. This work is partially funded by the EU: Requirements driven Software Development System (ReDSeeDS) (contract no. IST-2006-33596 under 6FP). The authors would like to thank ReDSeeDS partners for valuable discussions.

References

1. Kaindl, H., Smiałek, M., Svetinovic, D., et al.: Requirements specification language definition. Project Deliverable D2.4.1, ReDSeeDS Project (2007), http://www.redseeds.eu
2. Smialek, M., Bojarski, J., Nowakowski, W., et al.: Complementary use case scenario representations based on domain vocabularies. In: Engels, G., Opdyke, B., Schmidt, D.C., Weil, F. (eds.) MODELS 2007. LNCS, vol. 4735, pp. 544–558. Springer, Heidelberg (2007)
3. Requirements Driven Software Development System (ReDSeeDS) project. EU 6th framework IST project (IST-33596), http://www.redseeds.eu
4. Miller, J., Mukerji, J., et al.: MDA Guide Version 1.0.1, omg/03-06-01. OMG (2003)
5. OMG, Unified Modeling Language: Superstructure, version 2.2, formal/09-02-02 (2009)
6. Larman, C.: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice-Hall, Englewood Cliffs (2004)
7. Kalnins, A., Barzdins, J., Celms, E.: Model Transformation Language MOLA. In: Aßmann, U., Aksit, M., Rensink, A. (eds.) MDAFA 2003. LNCS, vol. 3599, pp. 62–76. Springer, Heidelberg (2005)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object Oriented Software. Addison-Wesley, Reading (1995)
9. Rein, M., Ambroziewicz, A., Bojarski, J., et al.: Initial ReDSeeDS Prototype. Project Deliverable D5.4.1, ReDSeeDS Project (2008), http://www.redseeds.eu
10. UL IMCS, MOLA pages, http://mola.mii.lu.lv/
11. Leal, L., Pires, P., Campos, M.: Natural MDA: Controlled Natural Language for Action Specifications on Model Driven Development. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 551–568. Springer, Heidelberg (2006)
12. Leonardi, M.C., Mauco, M.V.: Integrating natural language oriented requirements models into MDA. In: Workshop on Requirements Engineering, WER, pp. 65–76 (2004)
13. Osis, J., Asnina, E., Grave, A.: Computation Independent Modeling within the MDA. In: ICSSTE 2007, pp. 22–34 (2007)

What Is CIM: An Information System Perspective

Marite Kirikova, Anita Finke, and Janis Grundspenkis

Riga Technical University, Latvia
marite.kirikova@cs.rtu.lv, anita.finke@rtu.lv,
janis.grundspenkis@cs.rtu.lv

Model driven architecture usually is represented by computation independent model, platform independent model, and platform specific model. It is assumed that transformations from computation independent model to platform specific model via platform independent model are possible. However, authors rarely discuss the contents of computation independent model, its validity and correspondence to the notion „computation independent". Philosophically, from the point of view of organizational information systems pure computation independent models do not exist, because information technology solutions are threading through the way people think in their task performance, decision making and information search strategies. Considering information as interpreted data it is possible to distinguish between two inter-related models in the upper level of model driven architecture, namely, human intelligence information processing model and artificial intelligence information processing model, which are titled depending on the substance of the object that interprets the data available and recognizable in the business domain.

1 Introduction

Model driven development (MDD) and model driven architecture (MDA) are notions that have become very popular in the area of software engineering and information systems development. MDA is usually represented by computation independent model (CIM) at the upper level of the model hierarchy, platform independent model (PIM), and platform specific model (PSM). MDA is a general framework, which can accommodate different models belonging to UML and non-UML [2] notations at different levels of model hierarchy, e.g., [1] and [2]. Regarding CIM there are two basic streams of suggestions of what is to be represented by CIM. One of the streams suggests that the business model is to be represented at this level [3]. Another stream points to CIM as model, which represents system requirements [4]. Some researchers position both models representing business knowledge and system requirements at the CIM level [5]. There are also approaches that practically do not consider CIM [6]. This diversity of viewpoints on the nature of CIM points to the necessity to analyze the role of this model in MDA. In this paper the analysis of the nature of CIM is done from the point of view of information systems development. The research question stated is "How can CIM be incorporated in the information systems development process?". To answer this question we analyze the main elements of information systems and separate software from other information systems components. By doing this, the exact role of CIM emerges quite transparently and it is possible to identify

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 169–176, 2010.

particular information systems development situations and corresponding uses of CIM in these situations.

The paper is structured as follows. We discuss different approaches to CIM in Section 2. In Section 3 the notion of information system is analyzed. In Section 3 a new interpretation of CIM as consisting of human intelligence information processing model and artificial intelligence information processing model is proposed. In Section 5 different information systems development situations are analyzed and methods of the use of the information processing models are discussed. Section 5 consists of brief conclusions.

2 Related Works

MDA is a framework which attracts numerous systems developers. In many cases they try to interpret their previous and current methods in terms of CIM, PIM and PSM as well as develop new methods in MDA framework [1]. Table 1 represents some of these interpretations.

Table 1. Interpretations of CIM and PIM

No	CIM	PIM	Ref.
1	Environment of the system and requirements for the system	Part of the complete specification that does not change from one platform to another	[7]
2	Business Model, Domain Model, Business Requirements	BPMN Model Independent of workflow engine, UML model independent of computing platform	[8]
3	Task Model and Use Case Model	Abstract Interface Model and User Model	[9]
4	Requirements Models	Analysis and Design Models	[10]
5	Knowledge about the problem domain	Client's requirements, System Requirements Specification, Use Case Model, and Conceptual Model	[5]
6	Semantics of business vocabulary and business rules	Production rule representation, Specific rule language, Rule interchange format	[11]
7	Communication contracts, Global policies, Failover, Infrastructure capabilities	Routing model, Communications model	[4]
8	Product Line Requirement Model	Template driven portal-type interface	[12]
9	Business systems	Information systems	[13]
10	Business Process Model; System Requirements Model (Visual Use Cases)	Four archetypes: Moment-Interval, Role, (Party, Place, Thing), Description	[14]
11	Enterprise Integration Model: Organization Model, Process Model; Data Model, System Model	Service Model	[15]
12	-	Model of the system	[6]

Table 1 is structured in four columns. The first column refers to the number of the interpretation. Column 2 shows what is considered as CIM in a particular interpretation. Column 3 shows what is considered as PIM. Column 4 shows the reference sources for the particular interpretation. Interpretation of PSM is not reflected in the table as the contents of the paper concern mainly upper levels of MDA.

3 Information and Information System

In Table 1 presented in Section 2 only Row 9 explicitly shows the concept "Information System" (IS). This leads to the conclusion that the role of information system is not yet well understood in the context of MDA. One of the reasons of overlooking the impact of MDA on the organizational information system could be the diversity of definitions and understanding of IS notion [16]. Figure 1 illustrates a variety of concepts that are used in different IS definitions amalgamated in [16]. These concepts show that IS concerns all levels of MDA.

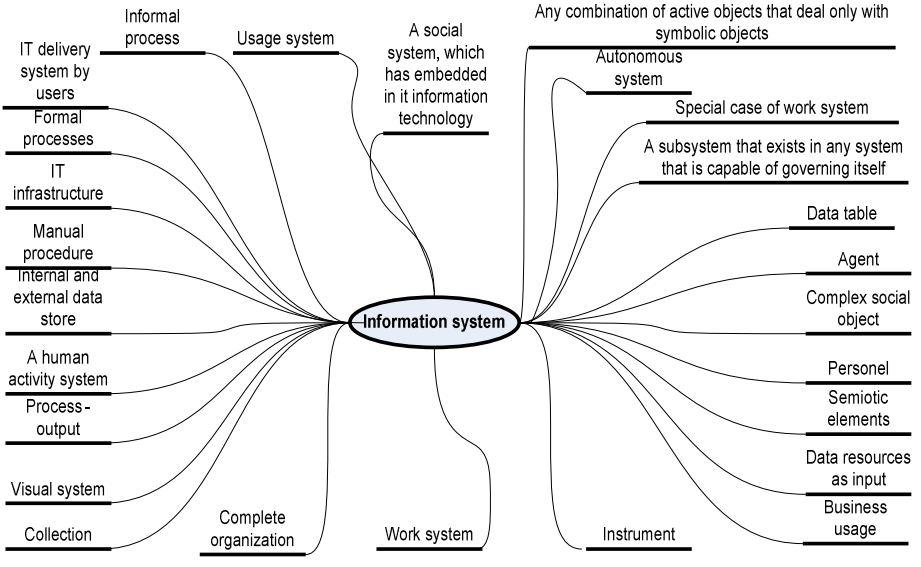


Fig. 1. Concepts used in information systems definitions

According to Steven Alter [16] "an IS is a work system whose processes and activities are devoted to processing information, that is, capturing, transmitting, storing, retrieving, manipulating, and displaying information. A work system is a system in which human participants and/or machines perform work (processes and activities) using information, technology and other resources to produce specific products and/or services for specific internal or external customers". In MDA the basic concern is software development. In fact, any new piece of software takes over from humans a particular information processing task which is a part of business level information

processing. The use of software usually changes the way how employees think when performing their routine tasks, because they have to understand at least partly what to expect of their "co-worker" - the software system. Therefore we can claim that philosophically there is no such phenomenon as computation independent model. In all kinds of enterprises where any tools for information processing are used the way of information processing (thinking) is more or less computation dependent. However in [7] it is stated: "The computation independent viewpoint focuses on the environment of the system, and the requirements for the system; the details of the structure and processing of the system are hidden or as yet undetermined". Taking into consideration that software essentially is a part of IS, the question arises what is *the environment* and what is *the system* from the point of view of IS in MDA. To answer this it is necessary to consider the relationship between human and software information processing.

In Knowledge Management literature [17] information is regarded as interpreted data. Thus for information to come into existence it is necessary to have some intelligent system that can interpret data. In general, the interpretation is the matter of human intelligence. But in software development situations, the software system imitates human knowledge interpretation at least at a conceptual level and at least to some extent. Therefore data received by software system from human performers is processed artificially according to data interpretation rules, patterns, and algorithms embedded in the system by programmers. Therefore one can consider CIM as consisting of two related abstractions:

- *Abstraction of human information processing*, which in this paper is called Human Intelligence Model of information processing (abbreviated as HIM)
- *Abstraction of software information processing*, which in this paper is called Artificial Intelligence Model of information processing (abbreviated as AIM)

HIM and AIM are connected by interface which represents data transfer from one model to another. An essential feature of data represented by the interface is its interpretability by HIM and AIM, so that the data would correspond to these information processing activities that are meaningful from the point of view of enterprise performance. The relationship between HIM and AIM is discussed in more detail in the next section. The interface belongs to both HIM and AIM. In MDA terms [7] HIM together with the interface corresponds to the environment and AIM together with the interface corresponds to the requirements for the system.

4 CIM = HIM Related to AIM

The question may arise concerning the level of detail HIM and AIM should be represented at. While the needed level of detail depends on the modeling situation and modeling purposes, there are the following essential features of the models and modeling situation that are always to be taken into the consideration:

- External input and output reachability
- Possibility of the identification of changes with respect to previous versions of models

In an organization the role of the IS is to support productive performance of the organization. Thus information flow has to reach decision points that are located on the border with organizational environment. The circulation of information is ensured by particular information processing steps that consist of input/output data interpretation aligned with information capturing, transmitting, storing, retrieving, manipulating, and displaying [16]. Three types of borderline decision points may be identified (1) external information input oriented, (2) external information output oriented, and (3) external information input and output oriented. When all relevant decision points are identified, it is necessary to decide about their mutual causal dependence. If the dependence exists, then there should be an information path between decision points. So the minimum level of detail of modeling requires possibility to identify all information paths between dependant borderline decision points. The paths may be located only in HIM, only in AIM, or in both models.

Nowadays organizations very rarely operate without using software for information processing. Thus the development of new software depends on the one that already exists in the organization [15]. If at the time point t_i the decision about a new software project is made, then the software should conform to HIM and AIM in time point t_{i+1}. AIM at time point t_{i+1} could be considered as requirements for software development; however, it is essential that the scope of the project is determined not only by AIM, but also by already existing software solutions. This leads to situation dependent methods of the use of AIM and HIM that are described in the next section.

5 Methods of the Use of AIM

Organizational information systems depend on ever changing business needs, hence the process of information systems development is a continuous sequence of implementation of different information systems projects including the development of new software. In approaches describing MDA, the main emphasis is usually put on the possibility to transform or map models of upper levels of MDA hierarchy to lower level ones, however it is not often that the models of human and artificial information processing are considered explicitly and separately, and taking into consideration time dimension.

Fig. 2 illustrates MDA in a continuous information systems development context. Traditional MDA thinking requires transferring/mapping CIM_{i+1} to PIM_{i+1} and PIM_{i+1} to PSM_{i+1}. However taking into consideration that during a particular project all software is rarely developed "from scratch" we have to consider the following basic modeling situations, each of which requires different methods of information systems development and use of AIM and HIM:

- *Situation 1*. The organization has no software; it is the first time when CIM_i and correspondingly HIM_i and AIM_i, (i=1 in Fig. 2) are built. Employees of the organization actively participate in requirements definition for the new information system. A part of their business knowledge is transferred into software solutions via AIM_i. Developing requirements they change their internal information processing models, and at the end of this process they understand what to expect from the new information systems solutions. If AIM_i is developed, it can be directly transformed into PIM_i (see Fig. 2).

- *Situation 2*. The organization already has software solutions. The as-is CIM (CIM_i in Fig. 2) and to-be CIM (CIM_{i+1} in Fig. 2) are being built. For new software solutions the conceptual gap ΔAIM between AIM_{i+1} and AIM_i is to be found. This gap then may be transferred into particular ΔPIM, it integrated to PIM_i will correspond to PIM_{i+1}. The question remains to what extent the new PIM_{i+1} corresponds to AIM_{i+1} and HIM_{i+1}, because the PSM_i is built on the basis of human knowledge existing at time point t_{i-1}. This knowledge can hardly be known and taken into consideration when building CIM_i and CIM_{i+1}.
- *Situation 3*. The organization already has software solutions. The as-is CIM (CIM_i in Fig. 2) exists or is being built and to-be CIM (CIM_{i+1} in Fig. 2) is defined. However, it appears that no new software is to be developed to map AIM_{i+1} into PIM_{i+1} and PIM_{i+1} into PSM_{i+1}. The software has to only be reconfigured. In this case the relationship between AIM_{i+1} and PIM_{i+1} is to be identified and documented to achieve the correct use of MDA. The question remains to what extent the new PIM_{i+1} (via AIM_{i+1}) corresponds to HIM_{i+1} because the PSM_i that is reconfigured to develop PSM_{i+1} that, in turn, corresponds to PIM_{i+1} is built on the basis of human knowledge existing at time point t_{i-1}. This knowledge can hardly be the same as the one reflected in CIM_{i+1}.

Considering the above described situations and assuming that CIM consists of HIM and AIM which are mutually related, it is clear that in each situation methods of use of AIM are different. The same applies to the use of HIM. Only in Situation 1 knowledge from HIM is directly transferred to AIM. In Situation 2 and Situation 3 HIM is quite alienated from AIM and transferability/mapping gap between HIM and AIM is to be analyzed to ensure professional use of software by users (e.g. by establishing proper training and educational programs). The width of the gap may depend on the involvement of users in the development of to-be models. User participation in to-be model development would allow expecting smaller transferring/mapping gap between HIM and AIM.

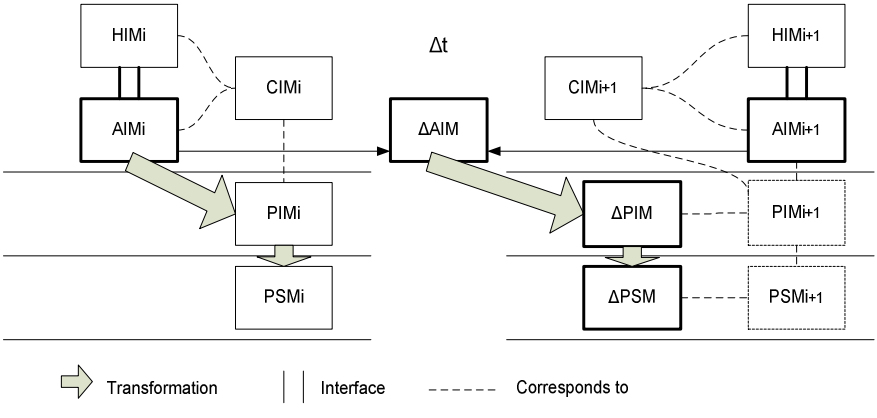


Fig. 2. MDA in continuous information systems development context

Above discussed situations do not cover all possible cases of software development. However they clearly show that from the perspective of IS development correct use of MDA approaches requires careful consideration of human knowledge at the CIM level. In Section 2 and Section 3 we suggested building CIM as a combination of HIM and AIM. In Table 1 (Section 2) practically none of the approaches addresses all relevant issues of HIM and AIM discussed in Section 4. On the other hand, the second column of Table 1 covers a wide area of methods used in organizational modeling and software engineering. The fact that none of them transparently and separably considers information processing by human and artificial intelligence suggests that new methods (which could take into consideration dependence of human thinking on software information processing) are to be developed for modeling at CIM level to achieve meaningful correspondence between HIM and AIM as well as correct transformations/mappings between AIM and PIM.

6 Conclusions

This paper analyzes MDA from the information systems perspective and proposes the position that models of how human and software "actors" collectively process the information are to be developed to achieve a correct use of MDA in IS development context. The following basic conclusions are drawn on the basis of the analysis of spectrum of IS definition elements and spectrum of interpretations of CIM:

1. It is useful to develop human and artificial intelligence information processing models at the upper level of MDA (HIM and AIM respectively).
2. It is necessary to ensure meaningful and right information interpretability between HIM and AIM.
3. External input and output reachability analysis helps to ensure completeness of CIM consisting of HIM and AIM.
4. It is necessary to take into consideration that correspondence between HIM and AIM may deteriorate with time due to the user alienated software development.
5. Different methods of the use of HIM and AIM are to be applied depending on the IS development situation.

Research findings presented in this paper are a part of a larger research on agile IS development and management. Currently, the methods for the representation and analysis of HIM aim AIM in the context of agile IS are under development.

Acknowledgment

The research presented in this paper is partly sponsored by Latvian Council of Science.

References

1. Fernandes, J.E., Mashado, R.J., Carvalho, J.A.: Model-driven development for perva-sive information systems. In: Mustefaoi, S., Mammar, Z., Giaglis, G. (eds.) Advances in Ubiquitous Computing: Future Paradigms and Directions, pp. 45–62 (2008)
2. ArcStyler MDA-Business For ArcStyler Version 3.x Transformer Tutorial (2002)

3. Becker, S.: Coupled model transformations. In: Proceedings of WOSP 2008, Princeton, New Jersey, USA, pp. 103–114. ACM, New York (2006), 978-1-59593-873-2/08/06
4. Biffl, S., Mordinyi, R., Scatten, A.: A Model-driven approach using explicit stakeholder quality requirement model for building dependable information systems. In: Proceedings of the Fifth International Workshop on Software Quality (WoSQ 2007). IEEE, Los Alamitos (2007), 0-7695-2959-3/07
5. Osis, J., Asnina, E., Grave, A.: Computation independent modeling within MDA. In: Proceedings of the 2007 IEEE International Conference on Software – Science, Technology and Engineering, pp. 22–34. IEEE, Los Alamitos (2007), 0-7695-3021-4/07
6. Guadalupe, O., Bordbar, B., Hernandez, J.: Evaluating the use of AOP and MDA in Web service development. In: The Proceedings of Third International Conference on Internet and Web Applications and Services, pp. 78–83. IEEE, Los Alamitos (2008)
7. Miller, J., Mukerji, J. (eds.): MDA Guide, Version 1.0.1 (2003)
8. Lankhorst, et al.: Enterprise Architecture at Work: Modelling, Communications and Analysis. Springer, Heidelberg (2005)
9. Valverde, F., Panach, I., Pastor, O.: An abstraction interaction model for MDA software production method. In: The 26th International Conference on Conceptual Modeling – ER 2007 – Tutorials, Posters, Panels and Industrial Contributions, Auckland, New Zealand (2007); Laender, A.H.F., Maciaszek, L., Roddik, J.F. (eds.) Conferences in Research and Practice in Information Technology, vol. 83 (2007)
10. Gherbi, T., Borne, I., Meslati, D.: MDE and mobile agents: another reflection on agent migration. In: Proceedings of UKSim 2009: 11th International Conference on Computer Modelling and Simulation, pp. 468–473. IEEE, Los Alamitos (2009), 978-0-7695-3593-7/09
11. Diouf, M., Musumbu, K., Maabout, S.: Methodological aspects of semantics enrichment in Model Driven Architecture. In: Proceedings of the Third International Conference on Internet and Web Applications and Services, pp. 205–210. IEEE, Los Alamitos (2008)
12. Kabanda, S., Adigun, M.: Extending Model Driven Architecture Benefits to Require-ments Engineering. In: The Proceedings of SAICSIT, pp. 22–30. University of Zululand (2006)
13. Slack, S.E.: The business analyst in model-driven architecture (2005), `http://www.ibm.com/developerworks/library/ar-bamda/index.html` (accessed, May 2009)
14. Kheraff, S., Lefebvre, E., Suryn, W.: Transformation from CIM to PIM using patterns and archetypes. In: Proceedings of 19th Australian Conference on Software Engineering, pp. 338–346. IEEE, Los Alamitos (2008), 1530-0803/08
15. Shuangxi, H., Yushun, F.: Model Driven and Service Oriented Enterprise Integration—The method, framework and platform. In: The Proceedings of the Sixth International Conference on Advanced Language Processing and Web Information Technology, pp. 504–509. IEEE, Los Alamitos (2007)
16. Alter, S.: Defining information systems as work systems: implications for the IS field. European Journal of Information Systems 17, 448–469 (2008)
17. Tiwana, A.: The knowledge management toolkit: Orchestrating IT, Strategy, and Knowledge Platforms, 2nd edn. Prentice Hall, Englewood Cliffs (2002)

Category Theoretic Integration Framework for Formal Notations in Model Driven Software Engineering

Gundars Alksnis

Department of Applied Computer Science, Riga Technical University, Meza iela 1/3, Riga, LV-1048, Latvia
gundars.alksnis@rtu.lv

Abstract. The paper presents research results of formal notations and Unified Modeling Language (UML) integration framework within the model driven software engineering. The originality of the solution is based on the combination of three technologies (formal notations, UML and category theory), in order to ensure their theoretical basis for integration in Model Driven Architecture. Framework is characterized by a principle according to which each UML model and corresponding formal specification is examined as an independent object, while morphisms between them specify how they are mutually linked. This allows to specify specific aspects of the system in a notation that is the best suited.

Keywords: MDA, UML, Category theory, Formal notations.

1 Introduction

Category theory (CT) is a field of mathematics that studies the properties of mathematical structures, by examining not the internal structure of the objects, but rather their relations (morphisms) with each other. CT can be used to identify interconnections between different unrelated structures (categories), and by using appropriate functors these structures can be linked, thus allowing to select the most appropriate structure for the solution.

The applications of CT in the computer science have at least two approaches. Classical approach also referred to as type theory or functional theory is perceived by computer scientists as a computer science focusing on computing [1]. Namely, categorical morphisms encapsulate the calculations or their abstractions, thus allowing to calculate target object from source object.

More recent CT approach examines the design of computer systems and is closely related with software engineering [2]. In this approach morphisms are regarded as relations between components, modules, software and the like artefacts of the system, for instance, refinement from high level specifications down to executable code.

In the paper we apply latter approach as a basis for formal notations integration framework with Unified Modeling Language (UML) in the model driven software engineering.

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 177–184, 2010.

2 Category Theory and Model Driven Architecture

Literature survey reveals that the scope of applications of CT in computer science is broad. However, there are several problems that have not been thoroughly analysed so far. One of them is nowadays topical issue of practical implementation of Model Driven Architecture (MDA) concepts [3].

On the one hand, MDA is based on such standards as UML, Meta Object Facility (MOF), Object Constraint Language (OCL), Query/ Views/ Transformations (QVT), Common Object Request Broker Architecture (CORBA) and others, that allow software designers to create and exchange models, as well as automatically generate software source code from them. On the other hand, majority of them are not yet completely formalized semantically. Therefore, there is a place for misinterpretations that impede the development of MDA tools.

One of the ways of eliminating ambiguity in the model elements is to introduce additional formal notations that would explain the system model in an unambiguous manner. We propose integration of formal notations with UML models and model elements, thus ensuring mutual complementation. UML model diagrams are mainly used to obtain easily perceivable views of the designed system, while formal notations define the systems formal specification in an unambiguous manner.

For this purpose, CT can be used to integrate formal notations and afterwards to integrate them in the development process of model driven software engineering. The use of CT for this solution is based on application of transformation approach – computing approach in this case is not applicable, since system models are perceived as objects and morphisms are perceived as relations between them, describing how one model acts as a part of another model or how one model is reflected in another.

By using CT constructions it is possible to define a framework that is independent from particular formal notations, as long as they are interlinked at the level of the same notation (e.g., metamodel). This means that each notation has own semantics; however there are semantic regions that are overlapping, such as illustrated in Fig. 1. Thanks to these overlapping regions, we can ensure mutual integration of the notations. The regions that do not overlap specify a certain system's viewpoint, for instance, data structure, process, modifications of system states over time etc., but all together they constitute a single system specification.

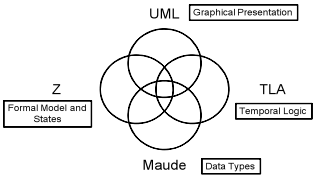


Fig. 1. Different formal notations semantic domains overlap scheme

3 UML and Formal Notation Integration Framework

In this chapter we sum up formal notation and UML model integration framework proposed and elaborated in [4], [5], and is partly inspired from [6].

When examining UML model transformations within MDA, it is necessary to define certain conditions and criteria based on which the transformation possibilities should be described and applied in practice.

Transformations in the context of MDA require to satisfy the following properties: defining of transformation rules, adaptation of transformations by means of parameters, ensuring of incremental consistency, two-way traceability, support for transformation objects, and classes and hierarchies of transformations [7]. We believe that as long as it is not possible to fully satisfy these conditions, the potential efficiency of transformation approach will not be used to the full extent and the ideas posed by MDA will not be fully implemented.

We propose the solution by the application of CT principles, since CT constructions have been used to formally describe each step of refinement and transformation in the specification of each model.

The main emphasis in the proposed framework is put on the practical application of CT, not its theoretical advancement. Namely, by using the studies of formal specification languages from the CT point of view, its constructions are applied to the specification integration and the analysis of semantics of the transformations.

The most important condition for formalisms to be integrated in MDA is the following: it must be possible to implement the UML and formalization of UML usage specification in easily expandable and automated manner, and the integration must support the composition of specifications, as well as the interoperability of the tools. Therefore, proposed framework focuses on metamodeling and the process of verification thereof, as well as the support of unified tools, which allows to perform formalization of semiformal modeling notations.

Another important aspect is that the framework should be based on MOF principles, i.e. UML metamodel should be taken into account, which, consequently, also requires formalization.

The general diagram of the UML and formal notations integration framework is illustrated in Fig. 2. The framework is abstract enough to be applied to any formal notation described below, but is not limited only to them.

Application of CT is manifested by the fact that the main emphasis is put on the relations between the elements of the framework nodes, namely, arrows. Namely, if each node is formalized in a category, arrows represent the relations between these categories (i.e. functors). The main difference between this framework and other frameworks of formalization is the following: formal specifications can be used to describe the structure and behaviour of objects, while CT based approach directly reflects the relations between the objects to be specified, namely, the emphasis is put on transformation relations.

In Fig. 2 arrows depict mappings (categorical functors) and have the following interpretation. Mapping (1) describes formalization of UML metamodel. Formal laws are used to describe how metaobjects of UML semantics should be

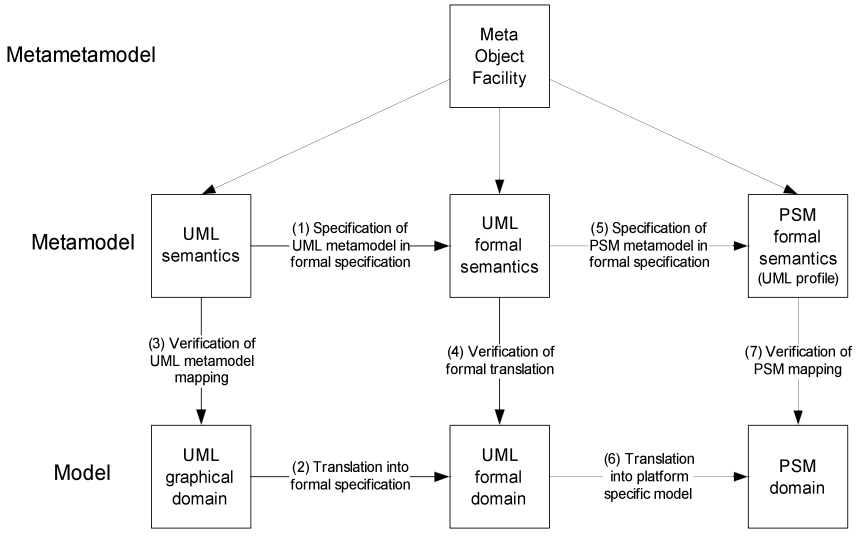


Fig. 2. UML and formal notation integration framework

transformed into formal specification (for instance, Z or Maude). In addition, the meaning of each transformation law and the possible alternatives of the transformation laws can be described.

Mapping (2) specifies the software and tool support requirements in order to automatically transform UML models into formal specifications in accordance with the formalization laws defined under mapping (1). The result is represented by formal specification or UML formal domain.

Mapping (3) specifies all the possible UML diagrams that can be created from the UML metamodel by using the tool. In other words, it is an interpretation of UML diagrams in UML metamodel. Mapping (1) and (3) jointly form the basis for further formalization of UML diagrams.

Mapping (4) ensures the verification of UML diagram translation. It helps to verify whether the formal specifications obtained from UML graphical domain correspond with the abstract UML theory of UML formal semantics.

Mapping (5) is an illustration of platform specific metamodel, which, for instance, could be one of the UML profiles or transformation models. This mapping also is described with the help of formal notation. Knowing the semantics of specification mappings and by using appropriate laws of transformations, with mapping (6) we can map each formal specification precisely and afterwards verify by using mapping (7) in accordance with the requirements of formal semantics of Platform Specific Model (PSM).

In this framework each specification module is examined as independent object, while morphisms between them demonstrate the mutual relations between specification objects. This allows specifying certain aspects of computer systems in the specification languages that are most efficient in terms of expressing these

aspects and properties (i.e., languages that are capable of describing required properties). For instance, one notation is used to describe states, while other is used to describe data structures and/or processes.

In case of CT approach the correctness of results in automated transformation of UML models into formal specification is reduced to the proof that the model class generated by transformer falls within the model class of UML metamodel. Thus it is necessary to prove the existence of morphisms from each UML metamodel theory to mapping theory of particular UML model.

We have defined metamodel level integration of UML state machines with Z notation as shown in Fig. 3 and UML class diagram integration with algebraic Maude approach (see Fig. 4).

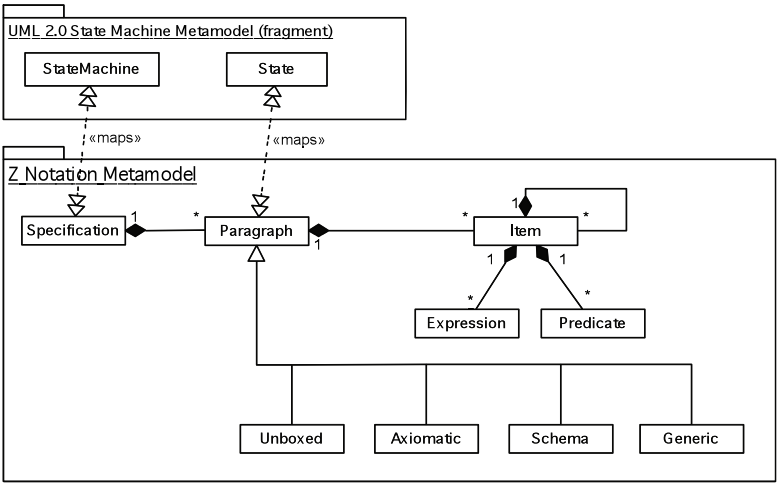


Fig. 3. UML state machine integration with Z notation

To sum up, Z specification corresponds to one state machine diagram, while each Z specification scheme corresponds to an UML state. Z specification can be used to specify invariant of state machine that is valid irrespective of the state and for each state it is possible to specify additional specifications in Z notation, which can be expressed as a scheme, axiom or general Z notation statement.

Algebraic Maude specifications are linked to the UML package, where each Maude object module is linked with particular UML class.

Verification of mappings can be automated. Namely, if both metamodel and model are formalized, the checking of mapping verification process is reduced to transformation of metamodel axioms into model theorems – whether the model is not conflicting with axioms defined within the metamodel. If conflicts are identified, they can be eliminated, since it is possible to determine, which particular model element is inconsistent.

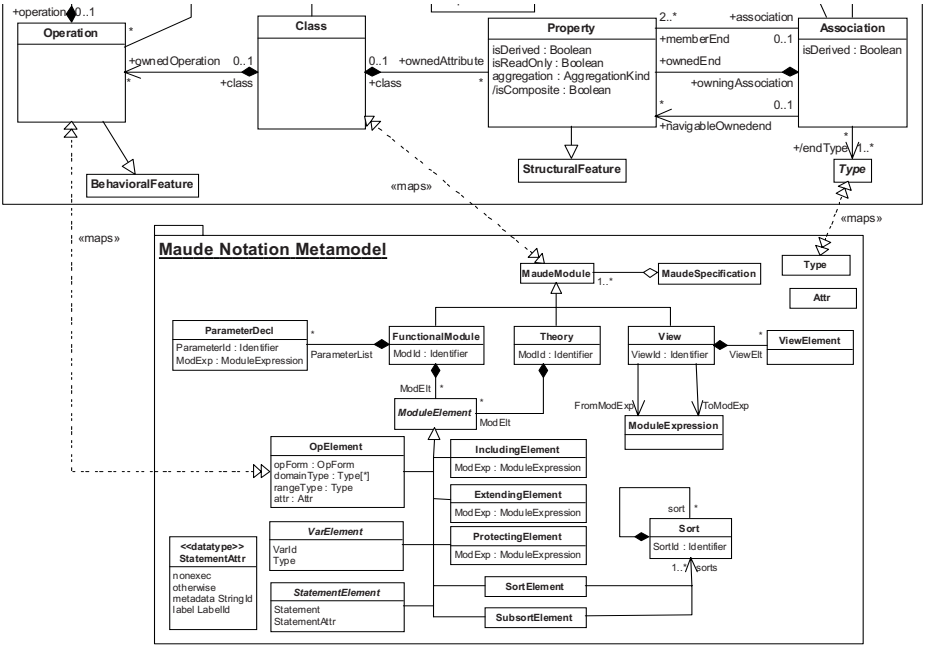


Fig. 4. UML class diagram integration with Maude notation

Model transformation into PSM is the next step after one has verified that Platform Independent Model (PIM) is correct form the point of view of UML formal syntax and semantics. The main aim of the PSM model is to ensure that it is possible to automatically transform it into executable code. Therefore, in this case it is not as important to clarify whether the model is semantically correct (given that the correctness is verified already in PIM), as to whether it is determined and can be translated to application code in an unambiguous manner. Namely, it is necessary to formalize each element of model up to the level that allows generating executable specification or code. In this case, it is necessary for each UML profile of PSM to formalize and demonstrate the relations of its concepts in a formal notation.

We have tested proposed framework in implementation of distributed transaction coordination system "TraSer" (from *Transaction Service*). It can be applied to support implementation of applications where possibilities of data locking offered by the data base management systems are not sufficient to efficiently implement client applications which operate at the business process level.

Algebraic Maude specifications of "TraSer" was specified to enrich systems static structure (i.e., to complement its UML class diagram), but translation from UML class diagram and state machine into Z specification was performed in order to formally specify the behavior of a system as reactive state machine. By linking each Z scheme of "TraSer" with appropriate state in UML state

machine model, a richer specification was obtained than in case of separate Z specification and UML state machine.

UML state machine defines easily perceivable yet informal aspect of systems dynamic properties, but Z notation supplements it by introducing formal specification of the model. For instance, Z notation specified system properties that are not directly represented in UML state machine, namely, one Z scheme specified the systems state scheme, i.e. systems invariant that had to be satisfied irrespective of the current state in UML state machine.

4 Related Work

As we already mentioned, the proposed framework is partly inspired from [6]. In it authors present metamodel based framework and demonstrate its application to UML model translations formal verification with the formal notation *Slang*. At time of publication their approach where not described in the context of MDA where PSM has to be introduced.

There are also other different attempts to describe MDA model transformations by means of UML metamodel. For example, in [8], authors propose metamodel for mapping specifications, but in [9] authors use pattern-based transformations.

Another integration method we inspired from is described in [10]. Its author proposes the method, which uses heterogeneous partial (also called viewpoint) specification integration. He also uses category theory to formalize specification languages and define a relational semantic framework.

We have also done research on UML model checking methods, where we outlined the main problems developer must deal with when UML state machine diagram model checking is performed [11].

The common conclusion is that the most notable shortcoming of current UML specification is the lack of its full formalization, which, in turn, makes tool creators do their own formalization, which may differ from other formalization approaches.

5 Conclusions

In order to correctly construct and maintain software systems, in the paper we present framework, which facilitates the integration of formal methods in the software engineering development processes where currently they are not used to the full extent. Studying of the possibilities of integration of formal methods and theories with partially formalized approaches may improve the methods and skills currently employed by the software development industry.

The originality of the proposed solution is based on the combination of three technologies (formal notations, unified modeling language and category theory), in order to ensure their integration in model driven software development process.

The proposed framework is characterized by a principle according to which each UML model and its formalized specification is examined as an independent object, while morphisms between them specify how they are mutually linked. It allows to specify certain aspects of system in a notation that is best suited to describe them. For instance, one specification examines the processes of a computer system, while other examines data structures and/or states. At the same time, category theoretical approach ensures well harmonized and justified theoretical basis for mapping of formal specification language structures and linking them via UML metamodel.

Combinations of both types of notations (informal graphical UML and formal mathematical notations) allow to obtain enriched PSM from which it is possible to generate more detailed application code. However, by applying further refinements it is possible to specify systems behaviour in more detail, in order to generate complete application code, the correctness of which can be verified by using the traceability information saved in the transformations and refinements.

References

1. Barr, M., Wells, C.: Category Theory for Computing Science, 3rd edn. Les Publications CRM, Montreal (1999)
2. Fiadeiro, J.L.: Categories for Software Engineering. Springer, Heidelberg (2005)
3. OMG Model Driven Architecture, http://www.omg.org/mda
4. Alksnis, G.: Formal Methods and Model Transformation Framework for MDA. In: Kolář, D., Meduna, A. (eds.) Proceedings of the 1st International Workshop on Formal Models (WFM 2006), pp. 87–94. MARQ, Ostrava (2006)
5. Alksnis, G.: Application of Category Theory to Integrate Formal Specification Languages in Model Driven Architecture. Summary of Doctoral Thesis. RTU Publishing, Riga (2008)
6. Smith, J., Kokar, M., Baclawski, K.: Formal Verification of UML Diagrams: A First Step Towards Code Generation. In: Evans, A., France, B.R., et al. (eds.) Workshop of the pUML-Group. LNI, vol. 7, pp. 224–240. GI, Toronto (2001)
7. Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture: Pratice and Promise. Addison-Wesley, Boston (2003)
8. Lopes, D., Hammoudi, S., Bézivin, J., Jouault, F.: Mapping Specification in MDA: From Theory to Practice. In: Konstantas, D., Bourrières, J.-P., et al. (eds.) Interoperability of Enterprise Software and Applications, pp. 253–264. Springer, London (2004)
9. Judson, S.R., France, R.B., Carver, D.L.: A metamodeling approach to model transformation. In: Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, pp. 326–327. ACM, New York (2003)
10. Bujorianu, M.C.: Integration of Specification Languages Using Viewpoints. In: Boiten, E.A., Derrick, J., Smith, G.P. (eds.) IFM 2004. LNCS, vol. 2999, pp. 421–440. Springer, Heidelberg (2004)
11. Alksnis, G.: The Analysis of UML State Machine Formal Checking Methods. In: Kelemenová, A., Kolář, D., et al. (eds.) Proceedings of the 10th International Conference Information System Implementation and Modeling (ISIM 2007), pp. 131–136. Silesian University, Opava (2007)

Application of BPMN Instead of GRAPES for Two-Hemisphere Model Driven Approach

Oksana Nikiforova and Natalja Pavlova

Riga Technical University, Kalku 1,
LV-1658 Riga, Latvia
{oksana.nikiforova,natalja.pavlova}@rtu.lv

Abstract. Models and model transformations are defined as primary artifacts in OMG's Model Driven Architecture (MDA). The main idea of MDA is to separate the formal representation of the system, preserving the highest possible level of abstraction in the form of system model, as well as to transform this model to the level required for system implementation. Two-hemisphere model represents the problem domain from the business perspective. It uses two inter- related system models—a business process model and a concept model, expressed in GRAPES notation. The main emphasis of this research is concerned with applying the OMG's notation to represent the business hemisphere in Business Process Modeling Notation (BPMN). This paper comprises the general issues of the research, difference between the notations, as well as the results on generated set of elements of UML class diagram.

Keywords: Two-hemisphere model, GRAPES, BPMN, model transformations.

1 Introduction

A model is an important artifact of software development and should represent the system to be developed from different aspects as close to the system as possible. The model provides information about the system, helps to find problems before implementation is started, and helps to find the solution of these problems. Model Driven Architecture (MDA) [1] is the central component in the strategy of Object Modeling Group to make the development process fast and qualitative. For achievement of this goal the role of explicit models and model transformations becomes more important.

Two-hemisphere model driven approach [2] is defined as a model driven approach for software development and satisfies the main statements of MDA on definition of models and model transformations. The essence of two-hemisphere model driven approach is to reflect the initial information about system in two main components – business process and concept models for further transformation to software. An investigation of model transformations defined for generation of UML class diagram [3] from two-hemisphere model is discussed in [4]. Two-hemisphere model [2] itself represents the problem domain and serves as a basis for further system development in the object-oriented manner [5]. Basically, the elements of two-hemisphere model are being transformed into elements of UML class diagram in a formal way [6]. Two- hemisphere model uses elements of GRAPES notation [7] for business process

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 185–192, 2010.

modeling and interrelated presentation of system concepts of problem domain. These two hemispheres are interrelated each to other by using GRADE tool [8], specified for GRAPES notation. The present paper continues author's investigations in the area of two-hemisphere model transformations under object-oriented software development and tries to find more suitable notation for presentation of functional hemisphere of problem domain. Business Process Modeling Notation (BPMN) [9] is developed under the supervision of Object Management Group (OMG) for acquisition and representing of initial business information. The fact that OMG is also an "ideologist" of Model Driven Architecture and its abilities for formal transformations of models gives to authors to make an assumption that BPMN can serve as a more suitable form for presentation of functional hemisphere of problem domain instead of GRAPES.

The goal of this paper is to research strengths and weaknesses to be detected under the change of notation from GRAPES into BPMN for business process model applied in two-hemisphere model driven approach for generation of UML class diagram. The comparison is made in relation to the class diagram, as since two-hemisphere model is used for generation of class diagram with its further generation into software code. The second section of this paper describes two-hemisphere model, applying of GRAPES and BPMN for business modeling in framework of two-hemisphere model. The third section describes advantages and disadvantages of applying of these two notations. The fourth section is conclusions of the paper.

2 Transformations from Two-Hemisphere Model into UML Class Diagram

The two-hemisphere model driven approach [2] is a version of business process model driven approach that utilizes two models of problem domain: the concept model and the business process model for driving the software development process in object-oriented manner. Fig. 1 shows the transformations between diagrams in accordance with two-hemisphere model driven approach [6].

As stated above, the two-hemisphere model is defined as two-interrelated models (Fig. 1): the first one is designed for system functionality (Process model), but the other one—for system concepts (Concept model). UML communication diagram is

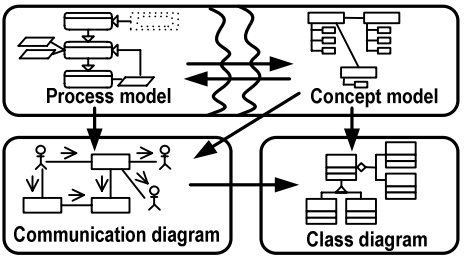


Fig. 1. Transformations from two-hemisphere model into UML class diagram

used to concatenate both hemispheres in one issue, as well as to share responsibilities of objects in object-oriented manner for software development. Then, UML class diagram is defined in accordance with the information of system operations in UML communication diagram; furthermore, the refinement of class attributes is done in Concept model. The formal algorithm for transformation from two-hemisphere model into class diagram is defined in [6]. Hence, [4] discusses several limitations on the set of elements of UML class diagram to be generated from two-hemisphere model expressed in GRAPES.

2.1 Two-Hemisphere Model Defined by GRAPES Notation

The original version of two-hemisphere model [2] contains two interrelated models: business process model (Process model on Fig. 1) and concept model (Concept model on Fig. 1.) defined in GRAPES [7] notation. Process model represents the functionality of the system in the form of processes, which are performed in order to achieve business goals, performers of these processes and events among the processes. According to Larman [10] real-world classes with attributes relevant to the problem domain and their relationships are presented in concept model. Concept model is a variation of well-known entity-relationship diagram notation. It consists of concepts (i.e. entities or objects) and attributes. Thus, elements of GRAPES notation used in two-hemisphere model are [7]:

— Business process usually means a chain of tasks that produce valuable result to some hypothetical customer, and is a gradually refined description of a business activity (task).

— Event is defined (as a rule) in the moment, when it is mentioned in business process diagrams for the first time. Events are the input/output objects of certain business process (perhaps, these can be material things or just information).

— Data object exists during one business transaction. From the viewpoint of information system specification, data objects should correspond to common files or variables: one task to fill in the information, another—to retrieve it.

— Performer section lists all objects involved in the completion of the task. The task may be performed by different sets of objects; in general, the possible per- formers are described by a logical expression with `&' and `|' connectors.

Fig. 2 shows an example for hotel room reservation system on how information from business process and Concept model is used for construction of UML communication diagram for its further transformation into class diagram. Tasks or processes, events, which executes transitions among tasks, as well as performers of tasks, defined in Process model (see in Fig. 2), are the main components for defining the UML communication diagram [3] with transformations defined in [6]. Objects of communication diagram are defined based on concepts of Concept model. Communication diagram can be defined directly from elements of business process model by using graph transformations [6]. The purpose of these transformations is concerned with the usage of processes (process model) as object operations (communication diagram), and events (process model) as owners and executors of operations (communication diagram). Class diagram, which is based on Concept model, is defined in accordance with information of object interaction (shown on communication diagram) [5].

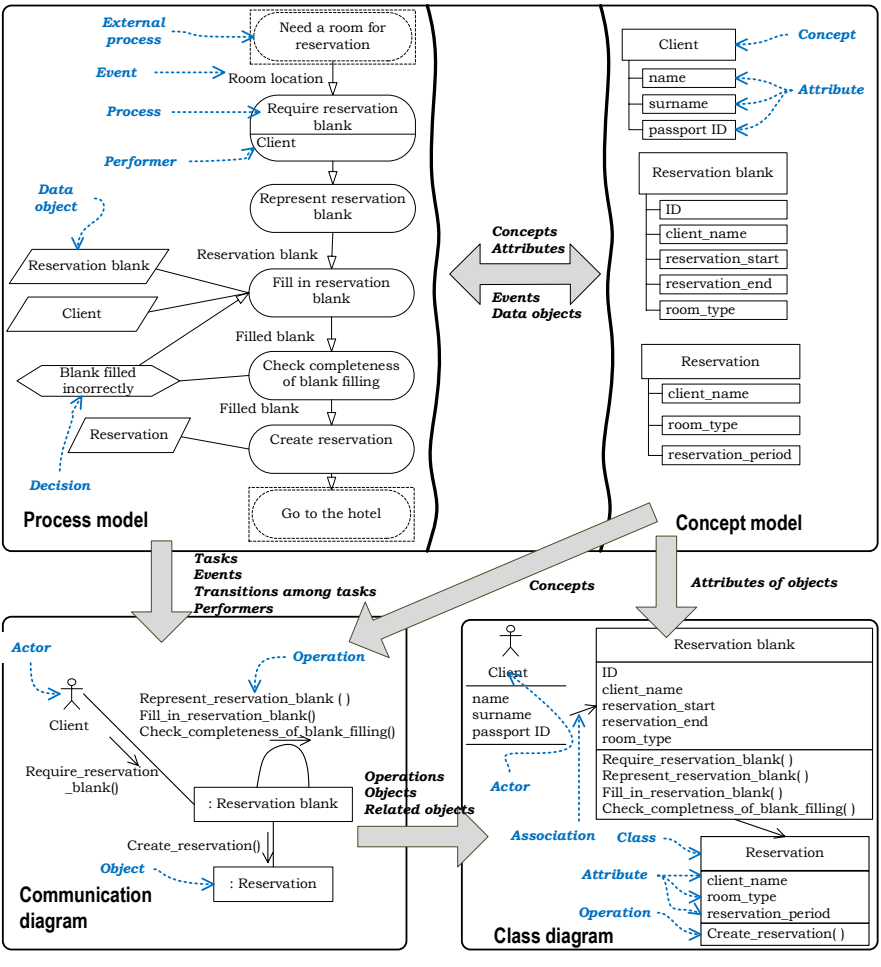


Fig. 2. Model transformations from two-hemisphere model defined in GRAPES notation

Furthermore, attributes of classes are defined just like the attributes of corresponding concepts in Concept model [6].

The investigation of 2HMD approach in [4] shows that approach could be applied for generation of several elements of class diagram, in order to generate class names, attributes, methods and relations between classes. Still, the ability to define stereotypes in classes for further definition of software architecture is missed in the two-hemisphere approach; also, the detailed definition of system dynamic component based on elements in two-hemisphere model is under development. In fact, these limitations are the motivation for further research, including the ability to set additional elements for two-hemisphere modeling of the system, to find more suitable notations for process model, and interrelated Concept model to generate fuller set of elements of UML class diagram.

2.2 Two-Hemisphere Model Defined by BPMN

Business Process Modeling Notation (BPMN) [9] is a standard for representing the business information [9]. Authors research the ability of the application of BPMN in order to represent the functional hemisphere in two-hemisphere model driven approach. Elements of process model in BPMN are defined as follows:

— Activity is a generic term for work that company performs. An activity can be atomic or non-atomic (compound). The types of activities that are a part of a Process Model are: Process, Sub-Process, and Task. Tasks and Sub-Processes are rounded rectangles. Processes are contained within a Pool [9].

— Pool represents a Participant in a Process, acts like a "swim-lane" and is a graphical container for partitioning a set of activities from other Pools [9].

— Data Objects are considered Artifacts as they do not have any direct effect on the Sequence Flow or Message Flow of the Process; however, they do provide information about what activities require to be performed and/or what they produce [9].

— Sequence flow shows the order of how the activities should be performed in a Process [9].

— Message flow shows the flow of messages between two participants that are prepared for information exchange (send/receive). In BPMN, two separate Pools in a Diagram will represent the two participants [9].

— Association associates information with Flow Objects. Text and graphical Non-Flow Objects can be associated with Flow Objects. An arrowhead on the Association indicates a direction of flow, when appropriate [9].

The transformations from process model (in BPMN) and interrelated Concept model has to be the same as the flow defined for process model in GRAPES notation. And the fact that central elements of process model in BPMN are activity (instead of process in GRAPES) and sequence or message flow (instead of event in GRAPES) make it possible to assume that the main statement of two- hemisphere modeling [6] has to be satisfied also in BPMN. That is the assumption that node of process model becomes an arc (operation to be fulfilled) in UML communication diagram and arc in process model becomes a node (object to interact) in UML communication diagram. Therefore the transformations for generation of elements of communication diagram are the same as defined for elements of two- hemisphere model in GRAPES notation (see an example in Fig. 3).

In addition, BPMN makes possible to generate new elements for object interaction and class diagram definition. Generally, data objects and concepts of Concept model are used to define the interacting objects. In case, with GRAPES notation all processes are transformed into operations of classes. However, in BPMN such elements allow to define the operations of objects on communication diagram; such elements as message flow and transitions between pools allow to judge about object of system interface, or so called classes-boundaries [3]. Additional element to the model transformations which are shown on Fig. 3, is class Form1—boundary class, which is defined in the place of transition among pools.

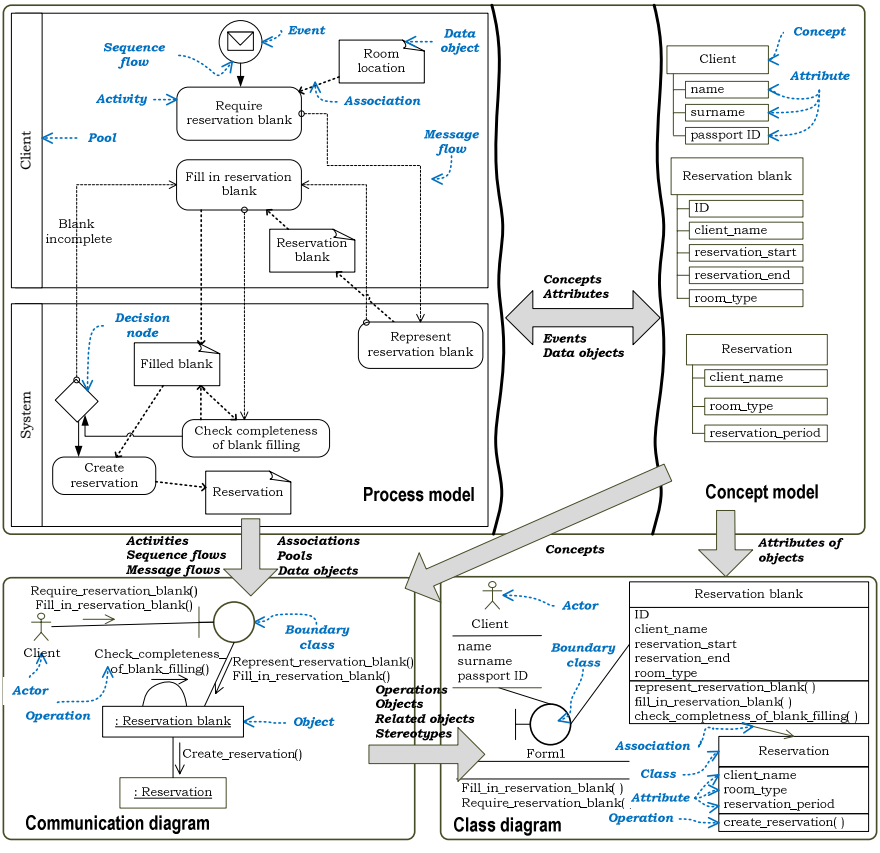


Fig. 3. Model transformations from two-hemisphere model defined in BPMN

Therefore, the application of BPMN for definition of functional hemisphere in problem domain refines generation of elements of class diagram with an ability to define stereotype class-boundary, which is a very useful component in object-oriented system development [3].

3 Several Outlines on Application of BPMN Elements Instead of GRAPES in Two-Hemisphere Model Driven Approach

Both notations may be used for representation of initial information and further system modeling in accordance with two-hemisphere system modeling. The strengths of GRAPES notation in comparison with BPMN are the following: GRAPES gives the ability to represent all of the elements required for business process model (support from GRADE tool is required [8]). This allows the creation of system model, its accordance with model completeness and consistency, and the generation of text description for elaboration. Thus, this ability is used in authors' research on possibility

to automate the transformations offered in two-hemisphere model driven approach. Tool for generation of class structure from two-hemisphere model is developed by authors and is described in [6]. This tool is based on textual description of business process and Concept model, generated by GRADE tool. However, this textual file is very specific and needs the standard description of its structure.

BPMN is originally proposed by OMG and is a standard notation in the business modeling domain. This notation is supported by many tools; it allows the generation of textual description for created business process model. Therefore, it is possible to continue the present research in the way of developing the tool to support the transformations defined by two-hemisphere model driven approach, where process hemisphere is expressed in elements of BPMN. The strengths of applying of BPMN are the following:

— BPMN separates the input/output data objects from transitions among processes. Thus, it becomes much easier to define the ownership of operations and all kinds of relationships among classes;

— In order to define area of responsibilities of every object, BPMN allows the representation of pools (this allow to judge about classes-boundaries);

— BPMN gives the ability to use an extension written in BPEL (Business Process Executive Language) [11], which is used to define a model that is transformable into executive code.

In general, BPMN has more notational details. Therefore, it gives the ability to define a more user-oriented model, than in GRAPES notation.

4 Conclusions

This paper summarizes the results of application of two notations for modeling of business information in two-hemisphere model driven approach. GRAPES is a business modeling notation used in the initial version of two-hemisphere model driven approach [2]. In turn, BPMN is a modern way to represent the business aspects of the system. Authors tried to investigate its abilities in generation of elements of UML diagrams in the context of two-hemisphere model driven approach instead of GRAPES notation for modeling of business hemisphere of problem domain.

According to research discussed in the paper the main difference between GRAPES and BPMN is that BPMN allows to model interaction of processes and separate the data objects from process flow, as well as to separate the modeling of the sequence flow from message flow. This gives a possibility to discuss about identification of operations from messages among the objects and definition of boundary stereotype for classes in object-oriented manner for development of the system. It could be chosen as directions of future work. Another future work direction is to find out the possibility of arguments of operations identification with detailed modeling of initial information – business process model.

Acknowledgments. The research reflected in the paper is supported by Grant of Latvian Council of Science No. 09.1245 “Methods, models and tools for developing and governance of agile information systems.”

References

1. MDA Guide Version 1.0.1, http://www.omg.org/docs/omg/03-05-01.pdf
2. Nikiforova, O., Kirikova, M.: Two-Hemisphere Model Driven Approach: Engineering Based Software Development. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 219–233. Springer, Heidelberg (2004)
3. Unified Modeling Language: Superstructure, version 2.2, OMG, http://www.omg.org/spec/UML/2.2/Superstructure/PDF
4. Nikiforova, O., Pavlova, N.: Open Work of Two-Hemisphere Model Transformation Definition into UML Class Diagram in the Context of MDA. Preprint of the Proceedings of the 3rd IFIP TC 2 Central and East Europe Conference on Software Engineering Techniques, CEE-SET (For internal use only), pp. 133–146 (2008)
5. Nikiforova, O.: General Framework for Object-Oriented Software Development Process. In: Scientific Proceedings of Riga Technical University. Series—Computer Science, Applied Computer Systems, Riga, vol. 13 (2002)
6. Nikiforova, O., Pavlova, N.: Development of the Tool for Generation of UML Class Diagram from Two-Hemisphere Model. In: Mannaert, H., Dini, C., Ohta, T., Pellerin, R. (eds.) Proceedings of The Third International Conference on Software Engineering Advances (ICSEA), International Workshop on Enterprise Information Systems (ENTISY), Conference Proceedings Services (CPS), Sliema, Malta, October 26-31, pp. 105–112. IEEE Computer Society, Los Alamitos (2008)
7. GRADE Business Modeling, Language Reference, INFOLOGISTIK GmbH (1998)
8. GRADE tools, GRADE Development Group, http://www.gradetools.com/
9. Object Management Group: Business Modeling Notation Specification. Version 1.1, http://www.omg.org/docs/formal/08-01-17.pdf
10. Larman, C.: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design. Prentice Hall, New Jersey (2000)
11. Business Process Execution Language for Web Services, Version 1.1, 2003-05-05, Copyright 2002, 2003, BEA, IBM, Microsoft, SAP AG and Siebel Systems (2003), http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/

Using Force-Based Graph Layout for Clustering of Relational Data

Vitaly Zabiniako

Riga Technical University, Institute of Applied Computer Systems,
1/3 Meza, Riga, LV-1048, Latvia
Vitaly.Zabiniako@inbox.lv

Abstract. Data clustering is essential problem in database technology – successful solutions in this field provide data storing and accessing optimizations, which yield better performance characteristics. Another advantage of clustering is in relation with ability to distinguish similar data patterns and semantically interconnected entities. This in turn is very valuable for data mining and knowledge discovery activities. Although many general clustering strategies and algorithms were developed in past years, this search is still far from end, as there are many potential implementation fields, each stating its own unique requirements. This paper describes data clustering based on original spatial partitioning of force-based graph layout, which provides natural way for data organization in relational databases. Practical usage of developed approach is demonstrated.

Keywords: clustering, database, relationship, force, graph.

1 Introduction

Clustering of objects is required in case if data units must be evaluated in terms of its similarity or semantic closeness. Two examples of such analysis in the field of database technology are optimization of storing data in memory and deriving hidden patterns in large loosely structured data sets.

The first case deals with necessity to decrease number of pages being retrieved while performing data querying [1]. Classic implementation of "Many-to-Many" relationship between two entities via auxiliary table is a good example – in case if according tables will be stored in different memory regions, unnecessary additional fetching of memory pages would be required (in comparison with mutually close storing of tables).

The second case is in relation with business analysis. Detecting of dense semantic relationships among certain entities (for example – consuming rates of products being developed and seasonal changes) may influence ongoing management strategy and even trigger development of new business rules crucial for increasing income. This also applies to exploration of different types of data, as in [2], [3], [4].

Considering that mentioned improvements might bring serious benefits in the context of work being done, there is no wonder that clustering problem became a major research topic in past years and continues to attract attention nowadays. Of course

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 193–201, 2010.

there is no universal clustering algorithm that would be able to make desired classification regardless semantic meaning of the data under inspection. That is why many clustering domains (i.e. hierarchical [5], partitional [6], spectral [7], etc.) emerged, each with its own set of algorithms and problem solving strategies. Detailed analysis of some existing approaches is presented in chapter 5.

In this paper the application of force-based graph layout calculation technique will be considered by author as initial step for clustering of relational data. It will be supplemented with custom partitioning strategy in order to provide full-fledged clustering algorithm.

Taking into consideration the above mentioned, it is possible to identify the following goal: to assist in databases design and analysis activities by providing force-based clustering mechanism. In order to reach this goal, there are being defined four subtasks: 1) to evaluate concept of force-based graph layout and outline its potential advantages for clustering; 2) to enhance this method with space partitioning strategy; 3) to evaluate results of such combination in a case study; 4) to make conclusion about benefits and drawbacks of force-based approach and its application in relation to other existing clustering methods.

2 General Characteristics of Force-Based Graph Layout and Its Clustering Capabilities

Force-based graph layout approach emerged in the field of graph visualization in mid 80-s, with such publications as Eades's "A heuristic for graph drawing" [8].

The main idea is as follows: geometric properties of elements of a graph are being calculated using simulation of forces. The graph is presented as a physical system, in which nodes are being replaced with metallic rings, while edges – with springs. At initial step rings are scattered randomly in space with deformation of springs. Taking into consideration, that springs will tend to compensate the deformation, all systems will also tend to find a state with minimal potential energy ξ.

This approach was originally defined in two-dimensional Euclidean space, although it might be also easily extended to the three-dimensional space, just by adding z (depth) component into position, velocity and kinetic energy calculations. There are many other potential improvements both of quantitative and qualitative characteristics of this approach, although in this paper it will be considered in its general form (for detailed description of mentioned improvements – refer to [9]).

The common emerging of graph layout pattern from initial random scattering is shown is Fig.1.

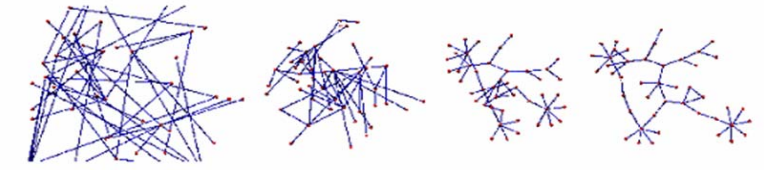


Fig. 1. Emerging of force-based graph pattern

The transformation of structure of relational database to corresponding graph is a straightforward process without any ambiguity – each table will be represented with unique graph node, while relationships between tables (according primary and secondary keys) will be captured in form of graph edges.

Fig. 1 depicts one secondary effect of force-based approach: the final layout tends to group densely interconnected nodes close to each other, while separating loosely connected groups in different space regions (to amplify this feature, repulsion force must express stronger effect than attraction force). This is a very useful property that highly correlates with clustering logics, although such positioning is not self-sufficient for classification. A space partitioning mechanism is required to derive the final desired results – separate sets containing unique data elements. The next chapter provides description of such mechanism along with its control parameters.

3 Space Partitioning and Regions Merging

The general logics of space partitioning require to divide the whole space under consideration into disjunction of separate isolated regions to identify mutually close objects within each region. The most common metrics of mutual "closeness" of two objects in Euclidean space is the geometrical distance *d*. Thus, the volume of an arbitrary region can be easily defined as a circle (in two-dimensional space) or a sphere (in three-dimensional space) which origin is equal to the center of this region and radius *d*.

Although this is the simplest way, it is not the most convenient for clustering – even the densest possible packing of spheres (without mutual penetration) produces gaps that may leave some nodes out of scope (marked with red) in Fig.2 part A.

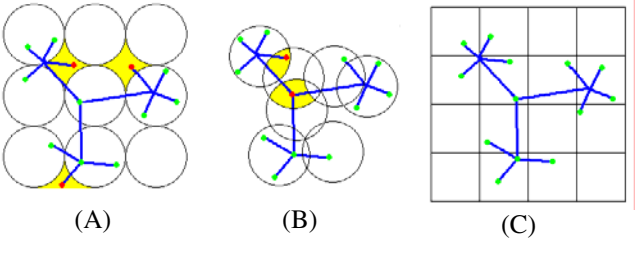


Fig. 2. Space partitioning using spheres versus orthogonal grid

Although allowing mutual penetration solves gap problem, it introduces another one – the need to resolve ownership in case if node traps in more than one region – see Fig.2 part B. The best way to overcome these problems, according to author's opinion, is the usage of an orthogonal grid, as it won't allow gaps or crossing of regions – see Fig.2 part C.

Additional advantage of orthogonal grid is in relation to performance (especially in case of multiple regions and nodes under inspection): queering whether an arbitrary node traps into an arbitrary region requires execution of a single "if" statement (testing along region bounding box planes), while determining of distance for a sphere involves calculation of a square root, which is relatively slower.

Two general models of space partitioning are being offered – two-dimensional grid model, based on recursive division of a square and three–dimensional grid model, using same approach for a cube. The number of regions being generated is controlled by a parameter *k* that defines how deep is the recursion division, as shown in Fig. 3.

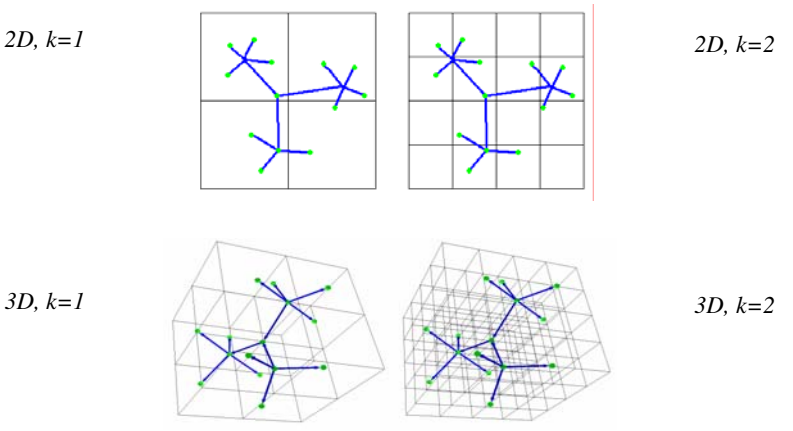


Fig. 3. Recursive space partitioning in two and three dimensions

The last mechanism needed for accomplishment of clustering is regions merging, otherwise (increasing *k*) we may end up with extreme situation, when each single node will be identified within its unique class. Merging of neighbor regions is the natural way how to bring multiple close nodes into corresponding cluster. A neighborhood of a region *R* is defined as a set of surrounding regions *{R'}* that share at least one edge or vertex with *R*.

Definition of a cluster with a set of neighbor regions is recursively transitive by its nature – region *R*, its neighbors *{R'}*, neighbors of neighbors *{R''}*, …, etc. belong to the same class. An author proposes the pseudo-code for the algorithm for assigning unique cluster identifications for an ordered set of regions that is as follows:

```
proc assign_cluster_id
   for each region r process_region(r);
end

proc process_region(r)
   if r = ∅ or r has id assigned then return;
   if r has neighbors with id assigned then
         assign same id to r;
   else
         assign new id to r;
   end
   for each neighbor n
       process_region(n);
   end
end
```

The first procedure *assign_cluster_id* initiates sequential processing of regions by calling subroutine *process_region*. When non-empty and un-marked region has been found, it becomes either a core of new cluster (generation of new identifiers is based on auto-increase principle) or a part of already existing one – depending on the state of surrounding regions. Then all its neighbors are recursively revisited.

Fig. 4 depicts example of sequence (left to right, top-down) of regions being investigated (marked with red numbers) and resulting clusters (marked with blue).

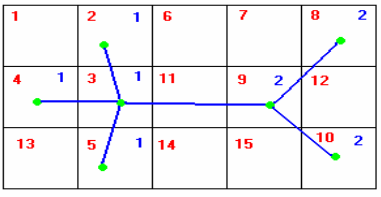


Fig. 4. Recursive space partitioning in two and three dimensions

4 Clustering Database Using Force-Based Method: A Case Study

In order to evaluate characteristics of application of force-based approach to relational data, modified *MS Access* template "*Northwind Traders Sample Database*" [10] will be used.

Original database consists of 8 tables, namely: "*Suppliers*", "*Categories*", "*Products*", "*Orders*", "*Order Details*", "*Employees*", "*Customers*" and "*Shippers*". In this article it is enhanced with 3 additional tables: "*Urgency*", "*Penalty*" and "*Reputation*". Related columns and mutual relationships between tables are shown in Fig. 5.

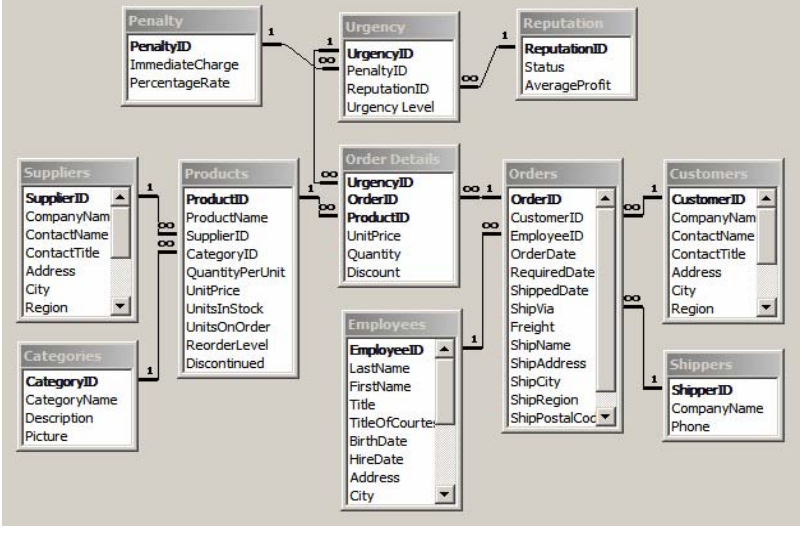


Fig. 5. Data relationships in modified "Northwind Traders Sample Database"

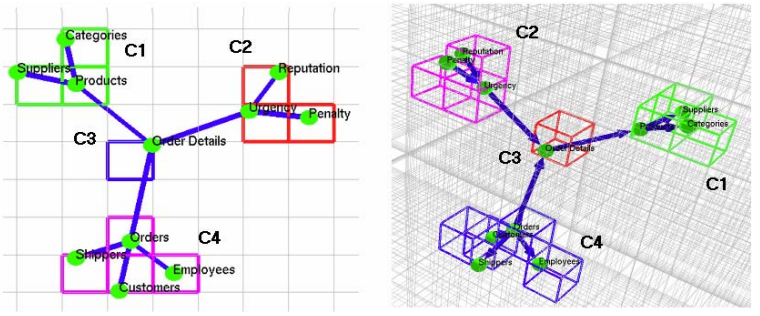


Fig. 6. Clustering of sample data

Common visual results of according graph clustering in 2D and 3D (with k=4 in both cases) are shown in Fig. 6.

In order to visually distinguish different clusters, thicker region borders and individual colors are being used. As it is seen, a set of four clusters is identified after running implementation of above-mentioned algorithm: C1={"*Suppliers*", "*Categories*", "*Products*"}; C2={"*Reputation*", "*Urgency*", "*Penalty*"}, C3={"*Order Details*"} and C4={"*Orders*", "*Employees*", "*Customers*", "*Shippers*"}. This is a meaningful solution that emphasizes semantic relationships between three separate domains – products, orders and urgency of completion of orders, via fourth logically interconnecting domain – product orders. As it was mentioned above, this information about data groups might be useful both for business analysis and data storing optimizations.

Although this certain result is logically valid, it is not the only one possible. Initial random scattering of nodes in a space (before the search of equilibrium state) produces a set of common clustering patterns. In this case these are as follows:

1. The one mentioned above.
2. C1={"*Suppliers*", "*Categories*", "*Products*", "*Order Details*"}; C2={"*Orders*", "*Employees*", "*Customers*", "*Shippers*"}; C3={"*Reputation*", "*Urgency*", "*Penalty*"}.
3. C1={"*Suppliers*", "*Categories*", "*Products*"}; C2={"*Orders*", "*Employees*", "*Customers*", "*Shippers*", "*Order Details*"}; C3={"*Reputation*", "*Urgency*", "*Penalty*"}.
4. C1={"*Suppliers*", "*Categories*", "*Products*"}; C2={"*Orders*", "*Employees*", "*Customers*", "*Shippers*"}; C3={"*Reputation*", "*Urgency*", "*Penalty*", "*Order Details*"}.

As it is seen, boundary domain "*Order Details*" can be trapped inside arbitrary primary group – the one that contains information about orders, products or urgency. This result is also valid, because the border table might be considered as logically related to all three primary domains equally (in case if no additional heuristics was provided by user).

In order to evaluate statistical distribution of these results, a set of experiments was carried out by author, to identify frequency of resulting patterns – refer to Table 1.

Table 1. Distribution of clustering pattern types

Two-dimensional model	**Pattern type**	**1**	**2**	**3**	**4**
	Count	23	10	6	11
	Frequency	0.46	0.2	0.12	0.22
Three-dimensional model	**Pattern type**	**1**	**2**	**3**	**4**
	Count	26	10	5	9
	Frequency	0.52	0.2	0.1	0.18

The graph of frequency distribution is shown in Fig.7.

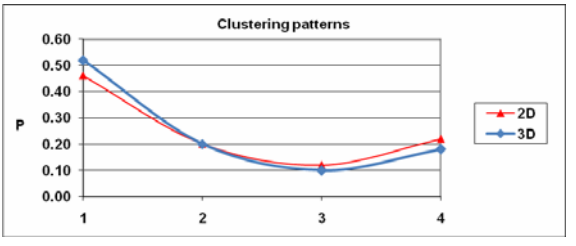


Fig. 7. Frequency distribution graph for pattern types

During statistical processing of clustering results, following metrics were acquired: mathematical expectation for two–dimensional model $M=2.1$, statistical dispersion $D=1.45$ and standard deviation $\sigma=1.2$. For the three-dimensional model, these values are following: $M=1.94$, $D=1.36$, $\sigma=1.17$.

It is possible to note that the most common pattern type is the first, while the third pattern is twice less as common as the second and the fourth. The explanation of this phenomenon is as follows: according to force-based layout calculation, "*Orders*" group consists of four elements and expresses stronger repulsion force, pushing the boundary table further to other groups (that consist of three elements and express weaker repulsion force). The attraction force in this case is equally compensated by all three primary groups and doesn't affect the position of boundary table.

5 Related Works

Different methods have been developed to solve clustering problems. Han, Karypis and Kumar [11] rely on hypergraph structure – according weighted graph is constructed to represent relations among different items in the original data array. This graph is then processed with "HMETIS" algorithm that minimizes the weighted hyperedge cut and produces partitions in which connectivity among the vertices is high. Resulting partitions are evaluated using fitness function that allows locating and eliminating bad clusters. Although the idea of assigning a set of related items to the single hypegraph edge is innovative, it requires additional modeling iteration.

Force-based approach requires no such overhead, as graph structure is captured in its original form from the input data.

The method proposed by Taskar, Segal and Koller [12] implies usage of probabilistic relational models (in its essence – templates for a probability distribution over a relational database of a given schema) for classification and clustering. According methods rely on extension of Bayesian networks to a relational setting. This, in turn, involves application of such techniques as "belief propagation" to evaluate mutual interconnection of elements. The choice of predetermined models requires careful selection based on the knowledge about the semantics of the underlying domains. This routine is similar to adjusting clustering parameters of force-based approach, however the latter requires no additional knowledge input.

Another approach by Yin, Han and Yu [13] which is called "CrossClus" performs cross-relational clustering with user's guidance. This algorithm accepts user queries that contain a target relation, and a small set of attributes. Then it searches for other pertinent features, and groups the target tuples based on those features. "CrossClus" adopts heuristic approach, which starts search from the user-specified feature, and then repeatedly searches for useful features in the neighborhood of existing features. In this way it gradually expands the search scope to related relations, but will not go deep in random directions. This concept is similar to proposed recursive space partitioning in force-based layout, although the latter has no predefined limiter for recursion level and examines all nodes local to a specific node under investigation.

Another important aspect of clustering approaches is in relation with performance. Nowadays O(n) complexity algorithms do exist, for example – refer to the algorithm introduced by Zong, Gui and Adjouadi [14]. Computation of graph layout with force-based method is relatively slow process that conforms to $O(n^2)$ complexity model. Still, this is not a critical task – clustering process must be initiated only after structural changes in table relationships. The most common tasks – changing or queering data has no impact on clustering because database structure stays the same.

Additional tempting benefit of proposed approach is as follows: in case if database structure needs to be visualized (for example to analyze it) there is no need to recalculate graph layout, as it is equal to the result of first clustering step. None of the other clustering methods provides this opportunity.

6 Conclusions

The general benefits of using force-based graph layout approach for clustering of relational data, according to author's opinion, are as follows: 1) wide choice of space partitioning strategies, merging models and its control parameters that makes this approach flexible enough to deal with different database relationship types; 2) intuitive light-weight clustering strategy that is based on natural way of treating densely interconnected elements by keeping them closely grouped.

The main drawback is as follows: proposed method is semi-automated – there is a need for imperative experimenting with a set of clustering parameters (such as minimal potential energy of equilibrium state ξ and depth of recursive space division k) in order to achieve best clustering results, otherwise final clustering may be unsatisfactory. For example, certain tables will trap in its own unique clusters regardless dense logical connections with nearby table groups.

Initial experiments with distribution of clustering patterns approved, that proposed approach may lead to valid logical results. Automatic distinguishing of logically desired solution (*M=1*) dominated in case of three-dimensional space: *M=1.94* versus *M=2.1* in case of two-dimensional model.

These are the main aspects of force-based clustering. The further investigation of this approach might include analysis of possible optimizations of force-based layout algorithm (together with the impact of according changes to clustering results), analysis of similar space division models (for example – based on hexagonal grid in two-dimensions and dodecahedrons in three-dimensions, etc.) as well as deeper analysis of distinctions of using 2D versus 3D space for clustering complex data.

References

1. Verant corp.: How Data Clustering Can Benefit Performance. A Versant Whitepaper (2007), http://www.versant.com/developer/resources/objectdatabase/whitepapers/vsnt_whitepaper_scalability_clustering.pdf
2. Férey, N., Gros, P.-E., Hérisson, J., Gherbi, R.: Visual data mining of genomic databases by immersive graph-based exploration. In: Proceedings of GRAPHITE 2005, pp. 143–146 (2005)
3. Koutsoudis, A., Arnaoutoglou, F., Pavlidis, G., Chamzas, C.: 3D content-based visualization of databases. In: Proceedings of the International Conference on Knowledge Generation, Communication and Management (2007)
4. Huang, M.L.: Information Visualization of Attributed Relational Data. In: Australian Symposium on Information Visualization, Sydney (December 2001); Conferences in Research and Practice in Information Technology, vol. 9 (2001)
5. Fung, B., Wang, K., Ester, M.: Hierarchical Document Clustering. Encyclopedia of Data Warehousing and Mining, 555–559 (2004)
6. Davidson, I., Wagstaff, K.L.: Measuring Constraint-Set Utility for Partitional Clustering Algorithms. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 115–126. Springer, Heidelberg (2006)
7. Neville, J., Adler, M.: Clustering Relational Data Using Attribute and Link Information. In: Proc. of the Text Mining and Link Analysis Workshop, IJCAI (2003)
8. Eades, P.: A Heuristic for Graph Drawing. Congressus Numerantium 42, 149–160 (1984)
9. Zabiniako, V., Rusakov, P.: Development and Implementation of Partial Hybrid Algorithm for Graphs Visualization. Scientific Proceedings of Riga Technical University, Computer Science, Applied Computer Systems, ser. 5, 34, 192–203 (2008)
10. Microsoft Corp., Northwind Traders Sample Database, http://www.microsoft.com/downloads
11. Han, E.-H., Karypis, G., Kumar, V.: Hypergraph based clustering in high-dimensional data sets: A summary of results. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 21, 15–22 (1997)
12. Taskar, B., Segal, E., Koller, D.: Probabilistic clustering in relational data. In: Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001), pp. 870–878 (2001)
13. Yin, X., Han, J., Yu, P.S.: Cross-relational clustering with user's guidance. In: KDD 2005: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 344–353. ACM Press, New York (2005)
14. Zong, N., Gui, F., Adjouadi, M.: A new clustering algorithm of large datasets with O(N) computational complexity, Intelligent Systems Design and Applications. In: Proceedings of 5th International Conference on Knowledge Discovery and Data Mining, pp. 79–82 (2005)

Data Structures for Multiversion Data Warehouse

Jan Chmiel

Poznań University of Technology, Institute of Computing Science
Poznań, Poland and QXL Poland (Allegro.pl)
jan.chmiel@allegro.pl

Abstract. In this paper we present data structures for the Multiversion Data Warehouse (MVDW). The data structures include: (1) *BitmapSharing* – developed for sharing data between multiple DW versions and (2) a MultiVersion Join Index – developed for supporting star queries that join multiple fact and dimension tables in the MVDW. The presented data structures have been evaluated experimentally showing their good performance.

Keywords: data structures, multiversion data warehouse, join index, star query, data sharing, multiversion join index.

1 Introduction

Nowadays, data warehouses (DWs) are indispensable in companies as core components of their decision support systems. From a technical point of view, a DW is a database that integrates data from various external data sources (EDSs). Data integrated in a DW are analyzed by the so-called On-Line Analytical Processing (OLAP) applications, for the purpose of discovering trends, patterns of behavior, anomalies, and dependencies between data. In order to support such kinds of analyses, a DW uses a dimensional model. In this model, an elementary information being the subject of analysis is called a *fact*. It contains numerical features, called *measures* that quantify the fact. Values of measures depend on a context set up by *dimensions* that often have a hierarchical structure, composed of levels.

Most of the existing technologies for data warehouses assume that data warehouse structures are time invariant. In practice, changes to these structures are frequent [28]. As a consequence, a DW structure evolves. One of the solutions to the DW evolution problem is a *MultiVersion Data Warehouse* (MVDW) approach [17,28]. It represents a schema and dimension evolution by the sequence of separate DW versions. Single DW version corresponds to, either the content of data warehouse within given time period, or to a simulation scenario. A DW version is composed of a schema version and an instance version that stores versions of data.

Versioning techniques similar to the MVDW have to solve an important problem of making the same set of data available for multiple DW versions. In a straightforward solution, the set of data that is used by multiple DW versions

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 202–210, 2010.

is physically copied into each of these DW versions, resulting in data redundancies, update anomalies, and additional system overhead for keeping the copies consistent.

Another research problem is related to processing queries in multiversion data warehouses. Typically, analytical queries intensively use *star queries* that join multiple dimension tables with a fact table. To this end, a special data structure, called a *join index* was developed [26] for traditional relational databases. Star query optimization in the MVDW is more difficult since data of user interest may be distributed among multiple DW versions and may be shared. As a consequence, the traditional join index cannot be directly applied to the MVDW. It has to be extended in order to support an efficient access to data that are physically stored in different DW versions and to data that are shared between multiple DW versions.

1.1 Contribution

This paper proposes solutions to the two research problems mentioned above. In order to solve the first problem we propose a technique called *BitmapSharing*. It was developed for sharing fact records and dimension records between multiple DW versions. In *BitmapSharing*, the information about all DW versions a given record is shared by is represented by the set of bitmaps, where one bitmap represents one DW version. The efficiency of the technique has been experimentally evaluated and compared to two advanced data sharing techniques proposed in the literature.

In order to solve the second problem we propose a data structure, called a MultiVersion Join Index (MVJI), developed for the purpose of indexing dimension and fact tables in the MVDW. Two kinds of the index were proposed, namely a ROWID-based one and a Bitmap-based one. Both kinds of the MVJI have been evaluated experimentally and compared to a straightforward approach where every DW version is indexed by an independent traditional join index.

2 Related Work

The work presented in this paper encompasses two research problems, namely: (1) data sharing techniques for multiversion databases/data warehouses, and (2) indexing versions of data.

2.1 Data Sharing

There are several techniques of sharing versions of data. From these techniques, the most advanced are *DBVA*, applied to object databases, and *Framework*, applied to relational databases. *DBVA* [2] uses the concept of a multiversion object. Each multiversion object contains a unique object identifier and the set of its versions. An object version can be either physical or logical. A physical version stores an object value whereas a logical version represents an existence

of a physical version in a given database version. Thus, multiple logical versions of an object may share the same physical version. In order to represent sharing, a physical version of object o_i has attached the set of database version identifiers that share o_i.

In *Framework* [22], records have associated sets of version ranges. A version range describes versions a given record is valid within. The version range contains a start version identifier and the set of end version identifiers. Each end version identifier points to the first version (in a version derivation graph) where a given record does not exist. The set of end version identifiers includes one identifier for one version derivation branch. In a database, records are stored as triples containing: sets of version ranges, unique key, and data.

2.2 Indexing Versions of Data

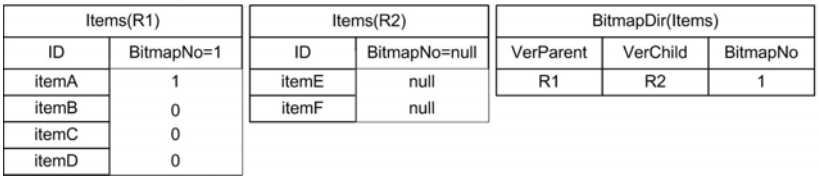
Several indexing techniques for the management of data versions were proposed in the research literature. Some of them focus on supporting access to data stored on write-once read-many media. The indexes are mostly B-tree based [6,13]. Other approaches focus on B-tree based indexes for managing temporal versions of data [8,12,1,27]. In [18,19] the authors proposed an indexing technique where time intervals (either valid or transaction) are mapped into a single value which is indexed by a B^+-tree. In [14,25] the authors proposed an index for indexing data records in a 2-dimensional space (transaction time and data value). In [22] the authors proposed an indexing technique for temporal versions of data records whose versions may branch. To this end, a B-tree like structure is used for indexing both data values and database versions, i.e. entries in index pages include: a key value and a database version. Physically, data records are partitioned into disk pages by version identifiers and keys. Recently, in [10], three different B^+-tree based index structures for multiversion data were compared analytically and experimentally.

The application of the aforementioned indexes to the MVDW is strongly limited since the indexes were developed for storing and searching versions of data that are stored in the same table. Secondly, they do not offer means for optimizing queries that join tables (possibly stored in multiple DW versions).

3 Sharing Data in Multiversion Data Warehouse

As mentioned before, sharing data between multiple DW versions is a desirable feature of the MVDW. To this end, we proposed a data sharing technique called *BitmapSharing* [5]. This technique enables sharing fact and dimension data by multiple DW versions. In this technique, information about DW versions that share a data record is stored with every record, in a fact or a dimension level table, by means of the set of bitmaps (bit vectors) attached to a shared table.

Figure 3 presents data sharing between 2 versions of fact table *Items*. Version *R1* of *Items* stores four records: *itemA*, *itemB*, *itemC*, and *itemD*. Version *R2* shares one record (*itemA*) with version *R1* and it contains two additional records,



Items(R1)	
ID	BitmapNo=1
itemA	1
itemB	0
itemC	0
itemD	0

Items(R2)	
ID	BitmapNo=null
itemE	null
itemF	null

BitmapDir(Items)		
VerParent	VerChild	BitmapNo
R1	R2	1

Fig. 1. The BitmapSharing technique

namely *itemE* and *itemF*. A bitmap describing data sharing between versions *R1* and *R2* is stored in *BitmapNo=1* attribute of table *Items(R1)*. The fact that bitmap number 1 describes data sharing between table *Items* in versions *R1* and *R2* is represented by record $<R1,R2,1>$ in additional data structure called *BitmapDir(Items)*.

BitmapSharing was evaluated experimentally and its performance was compared to *DBVA* and *Framework*. These techniques were implemented in Java. Data were stored out of a database in order to eliminate the influence of data caching and query optimization. The experiments were run on the MVDW composed of 10 linearly ordered versions. The experiments measured the performance of: constructing the content of a queried DW version, inserting data into a DW version, deriving a new DW version that shared data with its parent version, deleting records from a DW version.

Figure 3 shows the performance of constructing the content of a queried DW version. Two DW versions were queried, i.e. version no. 5 (located in the middle of the version derivation graph) and version number 10 (located at the end of the version derivation graph). Each version shared all its data with its parent version. The number of records physically stored (not shared) in each version was parameterized and equaled 10 000, 50 000, 100 000.

In this experiment, *BitmapSharing* offered the best performance. It is because for finding records belonging to a given DW version, the program had to retrieve bitmaps by executing simple table scans. Then, final selection of records was done by AND-ing the bitmaps. The number of shared records does not influence the processing time as the system processes the same number of bitmaps, regardless of the number of shared records. In the case of *Framework*, data were accessed by multiple reads of a B-tree index (efficient for queries that retrieve up to approximately 10% of records), which was not the case of the test scenario. In the case of *DBVA*, the sets of versions were read for all records and it caused a considerable time overhead.

Other experimental results (not presented here due to space limitation) show, that *BitmapSharing* outperforms *DBVA* and *Framework* for operations that include: (1) inserting data into a DW version, (2) deriving a new DW version that shares data. *BitmapSharing* offers worse performance for deleting data from a DW version. This feature is less important since usually, data are not deleted form a DW. They may be archived (moved from a DW into an external storage) but this operation is executed every a few years.

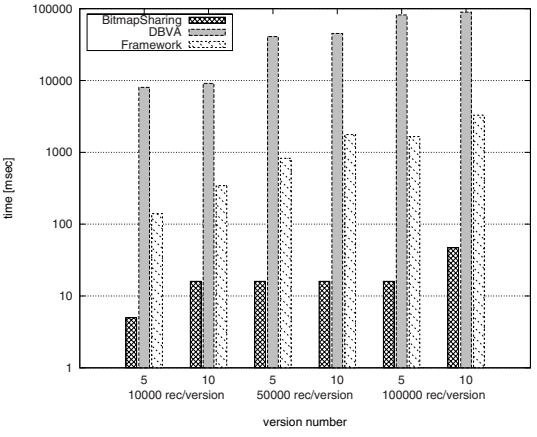


Fig. 2. Constructing the content of a DW version

4 MultiVersion Join Index

For the purpose of optimizing star queries in the MVDW, we propose the MultiVersion Join Index. The index joins multiple versions of a fact table with multiple versions of a dimension table. The index was developed in two variants, namely, a ROWID-based MultiVersion Join Index (R-MVJI) [3] and a Bitmap-based MultiVersion Join Index (B-MVJI).

4.1 R-MVJI

Its structure combines 2 indexes, namely *Value Index* (ValI) and *Version Index* (VerI). Both indexes are B^+-tree based. The structure of the R-MVJI is shown in Figure 4.1. The ValI is created on a join attribute, similarly as in a traditional join index. Its leaves store both: (1) values of an indexed attribute (denoted as K_1, K_2, ..., K_n) and (2) pointers to *VerI* (denoted as $VIptr_1$, $VIptr_2$, ..., $VIptr_n$). The VerI is used for indexing versions of a data warehouse. Its leaves store lists of ROWIDs, where ROWIDs in one list point to data records (of a fact and a dimension table) in one DW version. This way, for a searched value v of a join attribute A, the leaves of ValI point to all DW versions that store versions of records whose value of attribute A is v. A star query that addresses multiple DW versions can be answered with the support of the MVJI as follows. First, the ValI is accessed and searched for the value of a join attribute specified in the query. Second, being routed from the leaves of ValI, the VerI is searched in order to fetch versions of data records.

Since the R-MVJI is B-tree based, it offers good performance while defined on attributes of wide domains and for queries that access up to maximum 10% of rows. For attributes of narrow domains bitmap indexes are more efficient. For this reason, we developed also a Bitmap-based MVJI (B-MVJI).

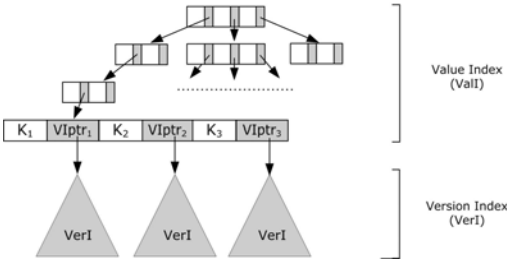


Fig. 3. ROWID-based MultiVersion Join Index

4.2 B-MVJI

The index is composed of two bitmap indexes, i.e., a *Value Bitmap Index* (ValBI) and a *Version Bitmap Index* (VerBI). The ValBI, created on a join attribute, points to records in all DW versions that have a given value of an index key. The VerBI points to records belonging to their DW versions. A query addressing fact and dimension tables in multiple DW versions uses the B-MVBJI as follows. First, the VerBI is accessed in order to compute a bitmap pointing to DW versions of interest. Second, the ValBI is accessed in order to compute a bitmap pointing to data records of interest. The final bitmap is computed by AND-ing the two aforementioned bitmaps.

4.3 Experimental Evaluation

The R-MVJI and B-MVJI were implemented in Java and evaluated experimentally. Their performance was related to a standard approach where each DW version contained its own standard join index. The experiments evaluated the performance of the indexes with respect to the number of versions accessed by a star query on multiple DW versions. The indexes were created on a join attribute whose selectivity equaled 0.1%. The values of this attribute were distributed evenly. Data records were stored in a file. Indexes of two different orders p={16, 32} were evaluated. The query selected 10% of data records from every DW version. The number of queried DW versions varied from 4 to 50. Every DW version contained 50 000 rows.

The experimental results are shown in Figure 4. The vertical axis represents the number of fetched index blocks, whereas the horizontal axis represents the number of DW versions accessed by a query. "nSJI" is the characteristic of the traditional approach, "R-MVJI" is the characteristic of the R-MVJI, and "B-MVJI" is the characteristic of the B-MVJI.

As we can observe from the charts, the B-MVJI performs better (requires less block accesses) for lower tree order p. For example, for p=16, the B-MVJI performs much better than its competitors when the number of DW versions accessed increases above 10. For p=32, the B-MVJI performs much better when

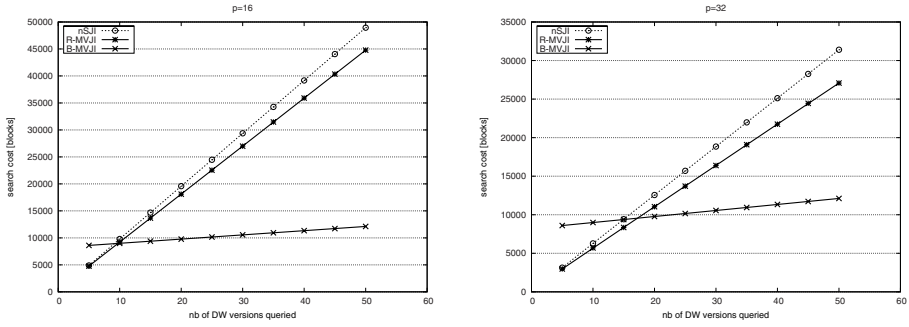


Fig. 4. Variable number of DW versions accessed by a star query

the number of queried DW versions is greater than 20. Such a behavior results from higher values of a B^+-tree height for smaller p. That, in turn, results in more B^+-tree pages that have to be fetched for indexes of smaller p. Bitmap indexes are not impacted by the value of p thus their performance characteristic remain constant.

5 Conclusions

This paper focused on data structures for the MultiVersion Data Warehouse. Two such structures were developed and evaluated experimentally. The first one – *BitmapSharing* was designed for sharing data between multiple DW versions. The second one – the MultiVersion Join Index was designed for supporting star queries that join multiple fact and dimension tables in multiple DW versions. The MVJI was developed in two variants, namely the ROWID-based MVJI and the Bitmap-based MVJI. The presented data structures have been evaluated experimentally showing their good (promissing) performance.

Currently we are developing a modification of the B-MVJI that includes bitmaps sorted by DW versions. Such an approach is acceptable since historical DW versions do not change their content, thus can be freely reorganized. Its pilot experimental evaluation shows that sorted bitmaps substantially improve query processing. In future we will develop an analytical cost model for the B-MVJI.

Concurrently we are also working on an index data structure suitable for indexing dimension hierarchies. Such a dimension index is applied to the optimization of analytical queries that perform roll-up and drill-down operations along a dimension hierarchy. First results are presented in [4]. Further research will focus on extending the concept of a dimensional index in order to encode a dimension taxonomy directly in the index.

References

1. Becker, B., Gschwind, S., Ohler, T., Seeger, B., Widmayer, P.: An asymptotically optimal multiversion B-tree. VLDB Journal 5(4), 264–275 (1996)
2. Cellary, W., Jomier, G.: Consistency of versions in object-oriented databases. In: Proc. of Int. Conference on Very Large Data Bases, pp. 432–441 (1990)
3. Chmiel, J., Morzy, T., Wrembel, R.: Multiversion join index for multiversion data warehouse. Information and Software Technology 51, 98–108 (2009)
4. Chmiel, J., Morzy, T., Wrembel, R.: HOBI: Hierarchically organized bitmap index for indexing dimensional data. In: DEXA (under review)
5. Chmiel, J., Wrembel, R.: Storing and sharing versions of data in multiversion data warehouse - implementation and experimental evaluation. Foundations of Computing and Decision Sciences Journal 32(2), 87–109 (2007)
6. Easton, M.: Key-sequence data sets on indelible storage. IBM Journal on Research and Development 30(3), 230–241 (1986)
7. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems, 3rd edn. Addison-Wesley, Reading (2000)
8. Elmasri, R., Wuu, G., Kim, Y.J.: Efficient implementation of techniques for the time index. In: Proc. of Int. Conf. on Data Engineering, pp. 102–111 (1991)
9. Johnson, T., Sasha, D.: The performance of current B-tree algorithms. ACM Transactions on Database Systems 18(1), 51–101 (1993)
10. Jouini, K., Jomier, G.: Indexing multiversion databases. In: Proc. of ACM Conference on Information and Knowledge Management, pp. 915–918 (2007)
11. Kolovson, C., Stonebreaker, M.: Indexing techniques for historical databases. In: Proc. of Int. Conference on Data Engineering, pp. 127–137 (1989)
12. Lanka, S., Mays, E.: Fully persistent B^+-trees. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 426–435 (1991)
13. Lomet, D., Salzberg, B.: Access methods for multiversion data. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 315–324 (1989)
14. Manolopoulos, Y., Kapetanakis, G.: Overlapping B^+-trees for temporal data. In: Proc. of Jerusalem Conference on Information Technology, pp. 491–498 (1990)
15. Morzy, M.: Advanced database structure for efficient association rule mining. PhD thesis, Poznań University of Technology, Institute of Computing Science (2004)
16. Morzy, M., Morzy, T., Nanopoulos, A., Manolopoulos, Y.: Hierarchical bitmap index: An efficient and scalable indexing technique for set-valued attributes. In: Kalinichenko, L.A., Manthey, R., Thalheim, B., Wloka, U. (eds.) ADBIS 2003. LNCS, vol. 2798, pp. 236–252. Springer, Heidelberg (2003)
17. Morzy, T., Wrembel, R.: On querying versions of multiversion data warehouse. In: Proc. of ACM Int. Works. on Data Warehousing and OLAP, pp. 92–101 (2004)
18. Nascimento, M.A.: A two-stage B^+-tree based approach to index transaction time. In: Proc. of Int. Workshop on Issues and Applications of Database Technology, pp. 513–520 (1998)
19. Nascimento, M.A., Dunham, M.H.: Indexing valid time databases via B^+-trees. IEEE Transactions on Knowledge and Data Engineering 11(6), 929–947 (1999)
20. O'Neil, P.: Model 204 architecture and performance. In: Gawlick, D., Reuter, A., Haynie, M. (eds.) HPTS 1987. LNCS, vol. 359, pp. 40–59. Springer, Heidelberg (1989)
21. O'Neil, P., Quass, D.: Improved query performance with variant indexes. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 38–49 (1997)

22. Salzberg, B., Jiang, L., Lomet, D., Barrena, M., Shan, J., Kanoulas, E.: A framework for access methods for versioned data. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 730–747. Springer, Heidelberg (2004)
23. Sinha, R.R., Mitra, S., Winslett, M.: Bitmap indexes for large scientific data sets: A case study. In: Parallel and Distributed Processing Symposium. IEEE, Los Alamitos (2006)
24. Sinha, R.R., Winslett, M.: Multi-resolution bitmap indexes for scientific data. ACM Transactions on Database Systems 32(3), 1–38 (2007)
25. Tzouramanis, T., Manolopoulos, Y., Lorentzos, N.A.: Overlapping B^+-trees: an implementation of a transaction time access method. Data & Knowledge Engineering 29(3), 381–404 (1999)
26. Valduriez, P.: Join indices. ACM Transactions on Database Systems 12(2), 218–246 (1987)
27. Varman, P., Verma, R.: An efficient multiversion access structure. IEEE Transactions on Knowledge and Data Engineering 3(9), 391–409 (1997)
28. Wrembel, R., Bębel, B.: Metadata management in a multiversion data warehouse. In: Spaccapietra, S., Atzeni, P., Fages, F., Hacid, M.-S., Kifer, M., Mylopoulos, J., Pernici, B., Shvaiko, P., Trujillo, J., Zaihrayeu, I. (eds.) Journal on Data Semantics VIII. LNCS, vol. 4380, pp. 118–157. Springer, Heidelberg (2007)

Location Based Information Storage and Dissemination in Vehicular Ad Hoc Networks

Girts Strazdins

Faculty of Computing, University of Latvia, 19 Raina Blvd., Riga, LV 1586, Latvia
gstrazdins@acm.org

Abstract. Vehicular Ad Hoc Networks (VANETs) is an emerging type of information networks in urban areas. A lot of research has been done in the area of increasing the vehicle awareness by disseminating collision and congestion warnings, and parking place availability information. In this paper we propose a novel idea and framework for dissemination of location based information, called digital maps, which are useful not only directly for the drivers and vehicle onboard navigation systems, but also external entities, such as tourists, environmental scientists, emergency services, advertisement companies. Centralized authority defines cooperative knowledge collection tasks and disseminates orders in the network while every vehicle decides which tasks it takes part of, based on hardware equipment, geographical position and individual interests of the driver. The results of preliminary simulation, with vehicles driving in an artificial city, show, that 200 vehicles/km^2 is minimum reasonable density to deploy proposed dissemination system.

Keywords: Distributed data bases, vehicular sensor networks, system architecture, framework.

1 Introduction

Vehicles are a significant part of everyday lives of urban people. Contemporary vehicles are equipped with onboard computers and wide range of sensors, measuring vehicles position, direction, speed, acceleration as well as various phenomena and events of surrounding environment: temperature, level of light, distance to obstacles and surrounding vehicles.

To improve the safety and efficiency of transportation, Intelligent Transportation System (ITS) paradigm is being intensively developed in recent years. The term ITS describes a range technologies with a common goal - to augment sensing capabilities and therefore enhance the intelligence of vehicles. To share the individual knowledge with other vehicles on the road, wireless vehicle-to-vehicle (V2V) communication support is added. Term *Vehicle-to-Infrastructure* (V2I) communication describes vehicles communicating with roadside units. DSRC and 802.11p (WAVE) are the state of art wireless vehicular communication protocols at the moment.

Using sensing and communication capabilities intelligent vehicular networks are created, on top of which distributed information systems are built. While each

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 211–219, 2010.

vehicle has limited resources and geographical coverage, together the network possesses a great potential for large scale information collection, storage and dissemination. In previous work this potential is mainly used for collision and congestion information exchange.

In this paper we propose a novel application for vehicular networks - collection of digital maps, containing location based information, which includes, but is not limited to: traffic flow speed and congestion; parking place availability; road quality; aggressiveness of drivers (breaking, acceleration, honks); temperature; air and noise pollution; weather indicators: raininess, fogginess, windiness.

Each of the tasks require specific sensing equipment, for example, thermometer, GPS. A vehicle can take a part of all the tasks it has required hardware for.

Our work has just began, the initial results of ongoing study are presented here. More extensive research and evaluation will be done, to find optimal system parameters and bring the proposed solution in deployment.

2 Related Work

Most of the previous research has been done in the area of collision detection and congestion information exchange. Authors of [1] propose to detect traffic jam events based on vehicle movement history. System called Nericell [2] explores the capabilities of identifying potholes, bumps and car horns on the city streets using mobile phone with accelerometers and microphone carried in a vehicle.

Physical limitations of wireless media have been studied in both simulations and real world experiments. Simulation results of the paper [3] show that maximum throughput is reached when 150-200 vehicles communicate simultaneously in the same area. A real world experiment from [4] concludes that communication time for vehicles on a highway varies from 15-30s while the average distance, at which the connection is established, is 133m. In [5] Torrent-Moreno et al. conclude that in saturated environments wireless broadcast reception rate can be as low as 20-30%.

Taking the above mentioned physical limitations in account, it is clear that it is not possible to use simple flooding for all the collected data. Various aggregation and diffusion techniques have been proposed in previous work. Authors of [6] propose to divide road in segments relative to each vehicle and aggregate data of each segment. In [7] passive on-demand cluster formation algorithm is advocated for data flooding in ad hoc networks. Authors of Cath-Up aggregation scheme [8] propose to forward messages with a random delay increasing the possibility of multiple reports to meet on their way and become aggregated. TrafficView [9] framework forwards messages in counter-flow direction, the same idea is used in [10].

Local data of multiple vehicles may appear contradictory, merging techniques must be applied to fuse it. Authors of [11] use modified Flajolet-Martin sketch as probabilistic approximation of real data values.

Pan hui et al. [12] propose to classify contacts based on their frequency and duration. The nodes with most frequent and long-lasting contacts form a community, having the greatest potential for data dissemination. Carreras et al. [13]

show, that knowing neighbor interests improves the possibility of finding required data in ad hoc networks.

Several prototype systems and frameworks for traffic data exchange have been built, including CarTel [14], TrafficView [9], SOTIS [15] and Mobile Century Project [16]. Authors of SOTIS prove the ability of their system to function even when a tiny part of all vehicles (2%) are equipped with it.

However none of the previous work has been done in the area of distributed digital maps with location based information in VANETs.

3 Problem Description

We examine tasks of collecting spatio-temporal data using vehicles. Let us denote a spatio-temporal map by a tuple $< c, w, h, u_W, u_H, t_F, t_T, t_U, V >$, where:

$c \in N \times N$: top-left corner of the map, described as geographical longitude and latitude, encoded with natural numbers;

$w, h \in N$: width and height of the maps in units u_W and u_H;

$u_W, u_H \in N$: size of one width/height unit of the map, in meters;

$t_F, t_T \in N$: time boundaries of the task (seconds after Jan/01/1970 00:00:00);

$t_U \in N$: time unit, described in seconds;

V: fact value set, map specific.

Example map:

$< (56862487, 24289398), 1000, 800, 10, 10, 1240203600, 1240581600, 60, \{0, 1\} >$

Here top left corner is a point near the city of Riga, the area is 10x8km, the collection task time boundaries are from 20/apr/2009 08:00 until 24/apr/2009 17:00, with a time unit of 60 seconds, and each fact can take value either 0 or 1.

The vehicular network contains a set of maps M. Each map contains a set of facts F. Let us denote fact as a tuple $< m, a, v, t >$, where

$m \in M$: the map;

$a \in N \times N$: coordinates of the fact, offset from top-left corner;

$v \in V$: value of the fact, map specific;

$t \in N$: time, when the fact was registered by its originator.

3.1 VANET Specifics

Vehicular ad hoc networks pose a set of specific problems:

1. Rapid changes in topology is the reason, why communication is the critical problem: links are very short and dynamic;
2. Vehicle on-board computers have not so strict resource limitations as conventional sensor network devices or mobile phones. Therefore more computation can be done to minimize and optimize the communication.
3. Vehicle movements follow a common pattern: streets are static. This fact can be used to predict future locations [17] and communication links.

4 Proposed Solution

The goal of our application is to merge local and limited information of individual vehicles to build a consolidated map containing the knowledge of the whole vehicular network. Advantages of our approach:

1. The map consists of numerous, exact local measurements with increased accuracy, compared to data collected by several, statically located devices (induction loops on the roads, meteorological stations).
2. The data accuracy increases in more dense regions, having more vehicles.
3. While a central authority is used to define the region and data to collect, and maintain security, it does not break the network scalability: vehicles act as both data collectors and aggregators. It is however advisable to have subset of network nodes connected to the central data base, giving opportunity to build a complete map and provide data for external users.

The potential users of the acquired data: vehicle drivers; tourists, city visitors and local inhabitants; public transport companies; public emergency services; environmental researchers; advertisement companies.

We argue that a completely centralized solution, where each vehicle sends the data and service requests directly to a central data base, is not reasonable because:

- Centralized solution is not scalable [18];
- Direct link with centralized server requires powerful and expensive infrastructure, and more energy is required for long range communication;
- Using cellular networks invokes cost per each transmitted byte.

There are, however, several advantages of central data processing:

- In situations, when city traffic is divided in separate clusters and data is never exchanged between them, having at least one node in each cluster connected to a central database, it would serve as data bridge;
- According to [19] it is unrealistic to assume, that V2V communication will function the same way in real world, as it does in simulations, because of the low V2V-ready vehicle percentage. Therefore even a minimal infrastructure improves the communication probability;
- More resources for data storage and report generation are available to a server compared to individual vehicles;
- To share collected data with third parties, a centralized sercer is necessary;
- To issue, sign and revoke security certificates for encryption and authentication, and define data collection tasks, a central trusted authority is needed.

Based on the above arguments, we propose a hybrid approach, where a central authority (CA) defines data collection tasks and maintains security while the data is disseminated in the network in a decentralized manner, using V2V communication. Subset of network nodes function as bridge-agents between the CA and decentralized network. Requirements, proposed architecture and system components are described in the following sections of this paper.

4.1 Requirements and Assumptions

To participate in the network, each vehicle is required to have:

- Global Positioning System (GPS) and navigation system with street maps;
- Radio communication unit (DSRC);
- Data processing unit;
- Graphical and/or audial user interface to interact with the driver.

The Intermediate Agents (IAs), which are described in the following section, require additionally a connection to the Internet, to communicate with both the vehicles (using DSRC) and the CA. The type of internet connection depends on the type of the IA - it could be a stationary base station on the road side, connected to a wired network, or a mobile taxi cab, which uses temporary connections while passing by WiFi access points. 3G cellular network is also a possible solution for IA connection to the server.

4.2 System Architecture

The distributed data network consists of nodes with the following roles:

- The Central Authority (CA) - a central server, which maintains a data base containing all the gathered data, defines data collection tasks, maintains security procedures and communicates with external data users. Connected to the Internet;
- Vehicles - responsible for data collection and dissemination in the network. Communicate with each other, using short range radio communication;
- Intermediate Agents (IA) - function as a bridge between vehicles and the CA, are connected to both vehicular network and the Internet. Deliver control meta data from the server into the network and transfer data between disconnected clusters of vehicles. A subset of vehicles (taxi cabs, public transport vehicles) or stationary base stations are potential players of the IA role;
- External data users - services and information systems which use the data provided by the CA.

4.3 System Components

System components are shown in Figure 1. They include three Input/Output devices, which communicate with the exterior: Radio, Sensors and User Preferences. *User Preferences* represent graphical/audial user interface providing communication with the driver of the vehicle. All other components are internal.

We do not specify exact algorithm for each component. For example, multiple aggregation techniques, used by Fact validator, are proposed previously, [11], [6] are some of them. Each component is a homogenous part of the system, using predefined I/O interface for communication with other system components. Multiple implementations are allowed for each component, switched at runtime.

The remainder of this section describes motivation and function of each component.

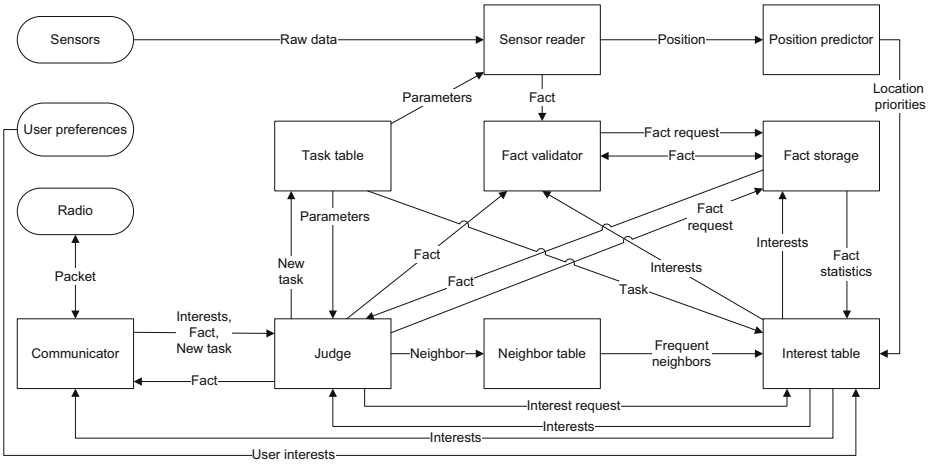


Fig. 1. System components

Communicator: transforms fact and interest objects into packets and vice versa. Exchanges packets with radio transmitter. Forwards received objects to Judge. Functions as data buffer and bandwidth controller. Performs aggregation techniques, which use data forward delay variation ([8]).

Judge: receives other vehicle interests and facts, as well as CA advertised new tasks from communicator. Answers and ignores requests, based on information received from Task table and Interest table. Also stores security information to check neighbor authenticity. Forwards data to Fact validator and Neighbor table.

Neighbor table: Stores actual and frequently visited neighbor list. Used for data muling: node also stores data, which it is not directly interested in, but which is requested by its neighbors [12], [13]. Modifies local interests by sending statistics to Interest table.

Interest table: Stores local interest information in form of (m, r, p), where $m \in M$ - the map, $r =< c_r, w_r, h_r >, c_r \in N \times N, w_r, h_r \in N$ - region, described by top-left corner coordinates c_r, width w_r and height h_r, $p \in N$ - priority. Makes decisions based on user input, actual and frequent neighbor interests, actual data collection tasks, statistics of actual stored facts (regions with more data become less interesting) and predicted future location.

Task table: stores actual data collection tasks. Receives new tasks, advertised by the CA, from Judge. Provides information to Sensor reader and interest table.

Sensor reader: Reads raw data from sensors and position from GPS, interprets it as facts, based on actual tasks. Forwards position to Position predictor.

Position predictor: Receives position of the vehicle and calculates statistics, which are used to predict future location [17], used to prioritize local interests.

Fact storage: Data base, storing actual knowledge of the vehicle in terms of facts - locally measured and remotely received. Sends facts to Fact validator and Judge upon request. Statistics are reported to Interest table.

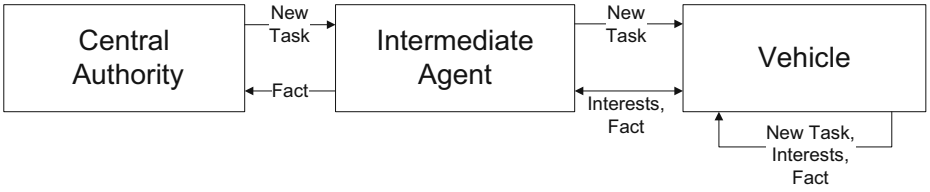


Fig. 2. Data flow in the network

Fact validator: validates new facts before storing them. Queries storage and performs aggregation when needed. Follows local interests. Serves as *non-validated data storage* [9]. Data validation takes computational resources and time, therefore this task has a low priority and is performed in background.

4.4 Communication Protocol

We identify the data flow between network nodes, shown in Figure 2, without specific implementation details, which depend on Communicator component used. Whether facts are broadcasted periodically ([8]), in specific directions ([9], [10]) or only sent in response to periodically broadcasted requests containing interests, is up to implementation.

Data flow contains three different objects:

New Task: request to start new data collection task. Created by the CA, sent to all IAs. Disseminated further in the network by each IA and vehicle. Must be signed by CAs private key to assure the authenticity of the tasks origin.

Interests: request sent by IAs and vehicles to let neighbors know facts of which regions should be sent first and in higher resolution. When sending fact reports, vehicles *should* take interests of local neighborhood into account.

Fact: data collected by vehicles. Each vehicle forwards its fact knowledge to neighbor vehicles and IAs. IAs act the same way as vehicles, but additionally they periodically forward received facts to CA.

Interest broadcasts or fact reports serve as beacons for neighbor tracking. Therefore smaller packets containing less interests and/or facts are preferred.

5 Evaluation

We have made a preliminary evaluation of vehicle communication patterns in a simulated city, using C++ simulation application. A city of size $1km \times 500m$ is used, streets forming a grid: horizontally and vertically, every 20×30 meters. Vehicles are placed on streets randomly, start driving in a random direction at a random speed, distributed evenly in interval 5-50 m/s. When crossing an intersecting street, a vehicle decides whether to go straight (8/16 chance), turn left (3/16), turn right (3/16) or turn around (2/16). When turning, vehicle chooses a new speed at which to travel until the next turn.

We run three different simulations with 10, 100 and 1000 vehicles respectively. Simulation time was chosen 10 minutes - enough to catch global trends. The

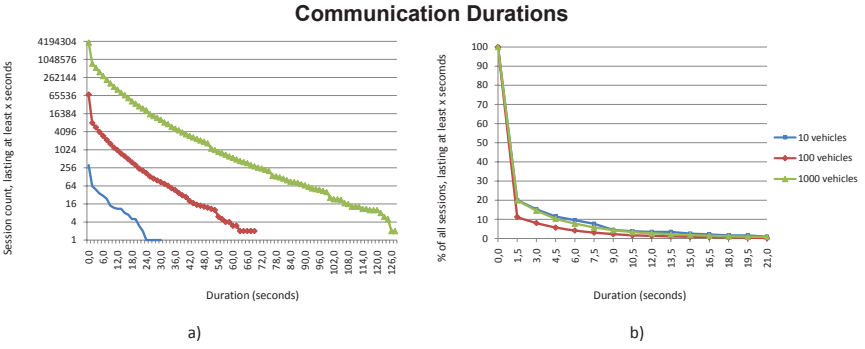


Fig. 3. Communication duration distribution: (a) count; (b) percent. Logarithmic scale.

communication time was fixed. Two cars are considered to be in a communication range, if they are on the same street (other area considered to be buildings, blocking signal) and the range between them is less than 200 meters (typical range for DSRC communication). The results are shown in Figure 3 - distribution of communication durations in terms of count and overall percentage.

Several conclusions can be made from the results. Communication possibility increases with vehicle count. However 80% of all sessions are shorter that 2 seconds, even for high vehicle density scenarios. Minimum reasonable vehicle density is at least $\frac{100}{1*0.5} = 200(vehicles/km^2)$.

6 Conclusion and Future Work

In this paper we propose a framework for location based information storage and dissemination in VANETs. Hybrid network architecture is used, having a central authority, which serves as centralized data base server, security maintainer and data provider to external users, and decentralized vehicular network for data collection and dissemination. Intermediate agents bridge the central authority and vehicular network together.

We have formalized the problem of data collection, described related work, potential advantages and drawbacks of both completely centralized and completely decentralized architectures. We describe the architecture of our proposed framework, identify data flow, system components and information flows in the network. A preliminary evaluation with simulated vehicular network in a small part of a city has been done. The results show, that even in dense scenarios 80% of communication sessions last only 2 seconds or less, therefore the data exchange must be done in an efficient manner.

In future work more extensive implementation and simulation work must be done. We will focus on simulating data exchange, using real life movement data traces from a set of real vehicles, driving around the city of Riga for multiple hours. A realistic radio signal propagation must also be used. Analysis of communication improvement by using static base stations is also planned.

Acknowledgements

I would like to thank my advisor Leo Selavo for the assistance, comments and suggestions related to this work.

References

1. Yoon, J., Noble, B., Liu, M.: Surface street traffic estimation. In: Proc. MobiSys 2007, pp. 220–232. ACM, New York (2007)
2. Mohan, P., Padmanabhan, V., Ramjee, R.: Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: Proc. SenSys 2008, pp. 323–336 (2008)
3. Eichler, S.: Performance evaluation of the IEEE 802.11 p WAVE communication standard. In: Proc. VTC 2007, pp. 2199–2203 (2007)
4. Seada, K.: Insights from a freeway car-to-car real-world experiment. In: Proc. WINTECH 2008, pp. 49–56 (2008)
5. Torrent-Moreno, M., Jiang, D., Hartenstein, H.: Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks. In: Proc VANET 2004, pp. 10–18 (2004)
6. Ibrahim, K., Weigle, M.: Accurate data aggregation for VANETs. In: Proc. VANET 2007, pp. 71–72 (2007)
7. Yi, Y., Gerla, M., Kwon, T.J.: Efficient flooding in ad hoc networks using on-demand (passive) cluster formation. Contract 14 [16] Mobile Century Project home page: http://traffic.berkeley.edu/
8. Yu, B., Gong, J., Xu, C.: Catch-Up: a data aggregation scheme for VANETs. In: Proc. VANET 2008, pp. 49–57 (2008)
9. Nadeem, T., Dashtinezhad, S., Liao, C., Iftode, L.: Trafficview: Traffic data dissemination using car-to-car communication. ACM SIGMOBILE Mobile Computing and Communications Review 8(3), 6–19 (2004)
10. Mitko Shopov, E.P.: Generation and dissemination of traffic information in vehicular ad-hoc networks. Computer Science (2005)
11. Lochert, C., Scheuermann, B., Mauve, M.: Probabilistic aggregation for data dissemination in VANETs. In: VANET 2007, pp. 1–8 (2007)
12. Hui, P., Yoneki, E., Chan, S., Crowcroft, J.: Distributed community detection in delay tolerant networks. In: Proc. MobiArch 2007 (2007)
13. Carreras, I., De Pellegrini, F., Miorandi, D., Tacconi, D., Chlamtac, I.: Why neighbourhood matters: interests-driven opportunistic data diffusion schemes. In: Proc. CHANTS 2008, pp. 81–88 (2008)
14. Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., Madden, S.: Cartel: a distributed mobile sensor computing system. In: Proc. SenSys 2004, pp. 125–138 (2006)
15. Wischoff, L., Ebner, A., Rohling, H., Lott, M., Halfmann, R.: SOTIS - A Self-Organizing Traffic Information System. In: Proc. VTC 2003, vol. 4 (2003)
16. :(April 2009)
17. Burbey, I., Martin, T.: Predicting future locations using prediction-by-partial-match. In: Proc. MELT 2008, pp. 1–6 (2008)
18. Wang, W., Lu, W., Mizuta, H., Wang, G., Du, Y., Liu, W., Tang, X.: Probe Car System based Traffic Information Service Experiment. In: Proc. IS Telecommunications, pp. 1134–1136 (2006)
19. Misener, J.: To 'V' Or Not To 'V'? Traffic Technology International, p. 12 (April/May 2009)

Software Development with the Emphasis on Topology

Uldis Donins

Department of Applied Computer Science, Institute of Applied Computer Systems,
Riga Technical University, Meza iela 1/3, Riga, LV 1048, Latvia
uldis.donins@cs.rtu.lv

Abstract. In this paper a problem domain and system static modeling formalization approach and formalization of static models based on topology borrowed from topological functioning model (TFM) is proposed. TFM uses mathematical foundations that holistically represent complete functionality of the problem and application domains. With the application of TFM within software development process it is possible to do formal analysis of a business system and in a formal way model static structure of the system. Software development starts with construction of TFM of a system functioning, after what the constructed TFM is transformed into problem domain object model. By doing further transformations of TFM it is possible to introduce more formalism in the Unified Modeling Language (UML) diagrams, their construction and in software development process. In this paper topology is introduced into the UML class diagrams and is defined approach for developing topological class diagrams.

Keywords: Problem domain modeling, software architecture, system static modeling, topological modeling.

1 Introduction

The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components. [3] and [8]

Since the publication of first UML specification, researchers have been working and proposing approaches for the UML formalization. Despite the fact that the latest UML specification [11] is based on the metamodeling approach, the UML metamodel gives information about abstract syntax of UML but does not deal with semantics which are expressed in natural language, and with the formalism of information contained in UML diagrams. The researchers deal mainly with formalization of UML syntax; formalization of information modeled with the UML diagrams is not their concern. For exampple, the aim of [2] is to work firmly in the context of the existing UML semantics: as a formalization instrument they use several formal notations, for example, Object Constraint Language or the formal language Z. Another research on

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 220–228, 2010.

formalizing UML constructs is [10] in which mathematical expressions are used to describe semantics of the class diagrams.

The main idea of the given work is to introduce more formalism into the UML diagrams and propose a formal approach for developing solution domain representing UML models. For this purpose formalism of a Topological Functioning Model (TFM) is used [6]. The TFM holistically represents a complete functionality of the system from the computation independent viewpoint. It considers problem domain information separate from the solution domain information. The TFM is an expressive and powerful instrument for a clear presentation and formal analysis of system functioning and the environment the system works within. Problem domain modeling and understanding should be the primary stage in the software development. This means that class diagrams must be applied as part of a technique, whose first activity is the construction of a well-defined problem domain model.

This paper is organized as follows. Section 2 describes the TFM development methodology and discusses the weak point of this methodology. Section 3 discusses the formalization of UML diagrams and sugested formalization approach for UML class diagrams which uses topology defined with the help of TFM. With the help of suggested approch it is possible to introduce more formalism into UML class diagrams by means of formalizing information conatined in these diagrams and precisely defining relations between classes. Section 4 shows an example of developing software by using suggested approach (software development with the emphasis on topology). Section 5 gives conclusions of this research and discuss future work.

2 Topological Functioning Model Development Methodology

TFM has strong mathematical basis and is represented in a form of a topological space (X, Θ), where X is a finite set of functional features of the system under consideration, and Θ is the topology that satisfies axioms of topological structures and is represented in a form of a directed graph. The necessary condition for constructing the topological space is a meaningful and exhaustive verbal, graphical, or mathematical system description. The adequacy of a model describing the functioning of a concrete system can be achieved by analyzing mathematical and functional properties of such abstract object [6].

A TFM has topological characteristics: connectedness, closure, neighborhood, and continuous mapping. Despite that any graph is included into combinatorial topology, not every graph is a topological functioning model. A directed graph becomes the TFM only when substantiation of functioning is added to the above mathematical substantiation. The latter is represented by functional characteristics: cause-effect relations, cycle structure, and inputs and outputs. It is acknowledged that every business and technical system is a subsystem of the environment. Besides that a common thing for all system (technical, business, or biological) functioning should be the main feedback, visualization of which is an oriented cycle. Therefore, it is stated that at least one directed closed loop must be present in every topological model of system functioning. It shows the “main” functionality that has a vital importance in the system’s life. Usually it is even an expanded hierarchy of cycles. Therefore, a proper cycle analysis is necessary in the TFM construction, because it enables careful analysis of system’s operation and communication with the environment [6].

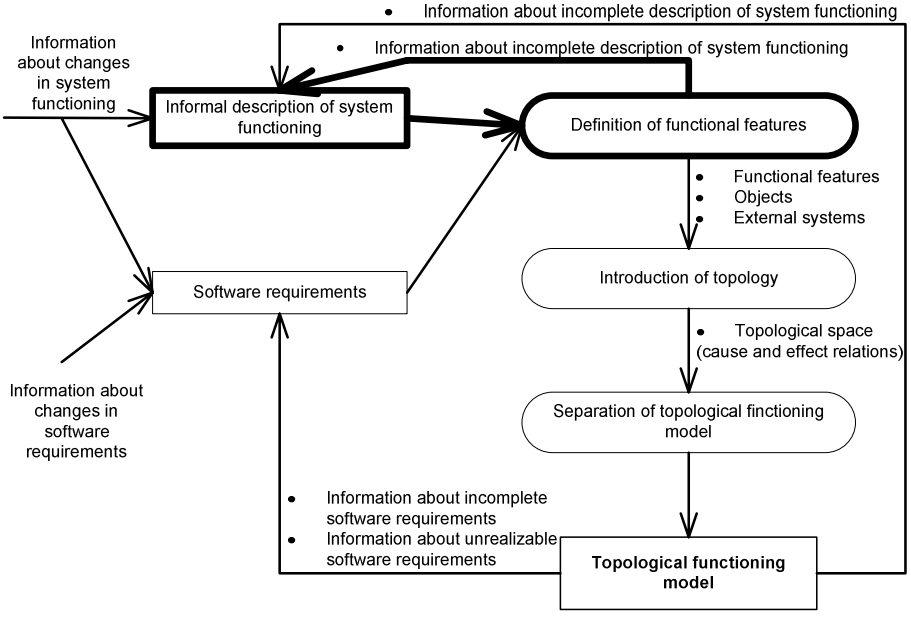


Fig. 1. Topological functioning model development process

Schematic representation of the development process of TFM is given in Fig. 1.

Within previous researches there are stated three steps for developing TFM of system functioning [6] [7]:

Step 1: Definition of physical or business functional characteristics, which consists of the following activities: 1) definition of objects and their properties from the problem domain description; 2) identification of external systems and partially-dependent systems; and 3) definition of functional features using verb analysis in the problem domain description, i.e., by finding meaningful verbs.

Step 2: Introduction of topology Θ (in other words – creation of topological space), which means establishing cause and effect relations between functional features. Cause-and-effect relations are represented as arcs of a directed graph that are oriented from a cause vertex to an effect vertex. Topological space is a system represented by Equation (1),

$$Z = N \cup M \tag{1}$$

where N is a set of inner system functional features and M is a set of functional features of other systems that interact with the system or of the system itself, which affect the external ones.

Step 3: Separation of the topological functioning model from the topological space of a problem domain, which is performed by applying the closure operation over a set of system's inner functional features (the set N) as it is shown by Equation (2),

$$X = [N] = \bigcup_{\eta=1}^{n} X_{\eta} \tag{2}$$

where Xη is an adherence point of the set N and capacity of X is the number n of adherence points of N. An adherence point of the set N is a point, whose each neighborhood includes at least one point from the set N. The neighborhood of a vertex x in a directed graph is the set of all vertices adjacent to x and the vertex x itself. It is assumed here that all vertices adjacent to x lie at the distance d=1 from x on ends of output arcs from x.

Construction of TFM can be iterative. Iterations are needed if the information collected for TFM development is incomplete or inconsistent or there have been introduced changes in system functioning or in software requirements.

While the three steps for construction of TFM is well defined, the method for creating sufficient informal description (the problem domain description) is not developed yet. When creating TFM of system's functioning a number of questions arise:

- How much information is needed within informal system description?
- Is it needed to identify roles which are responsible for doing corresponding actions?
- How much attributes of objects to include in informal system description?
- How to make structure of informal system description?
- And finally: when the informal description of system functioning is finished and sufficient for successful software system development?

These questions cover part of Fig. 1. (this part is denoted with bold lines).

3 Formalization of Unified Modeling Language

With the help of TFM it is possible to introduce more formalism in UML diagrams. The formalization level of UML diagrams is increased by transforming topology from TFM into UML diagrams' elements. There are several researches made for UML formalization, for example, [2] and [10]. Mainly researches for formalizing UML diagrams are concerned with formalization of UML diagrams' syntax and notation. This is because the UML specification [11] is given in natural language. Within this research formalization of UML diagrams is stated as formalization of information contained in UML diagrams. The idea about topological UML diagrams is published in [5].

As the first diagram in which to introduce more formalism by using topology is chosen class diagram. Class diagrams reflect the static structure of the system, and with the help of class diagrams it is possible to model objects and their methods involved in the system. Regardless of the opportunities provided by the class diagrams, it is not possible to reflect the cause and effect relation within a system or to indicate which certain activity accomplishment of an object triggers another object's certain activity accomplishment.

Topological relations between classes throughout this article are marked with directed arcs (this means that within this article notation used for topological relations between classes is similar to notation of associations in UML). The example of topological relations can be viewed in Fig. 2.

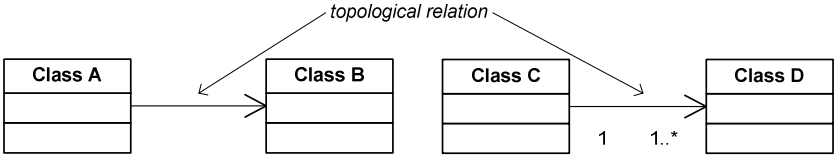


Fig. 2. Example of topological relations between classes

Before topological class construction it is needed to construct the TFM of the system functioning. After construction of TFM it is possible to transform topology defined in TFM into class diagrams. It is possible to transform topology from TFM into class diagrams because TFM has strong mathematical basis. In this way the formalism of class diagrams means that between classes are precisely defined relations which are identified from the problem domain with help of TFM. In traditional software development scenario relations (mostly associations and generalizations) between classes are defined by the modeler's discretion.

In order to develop a topological class diagram, after the creation of TFM a graph of problem domain objects must be developed and afterwards transformed into a class diagram. In order to obtain a problem domain object graph, it is necessary to detail each functional feature of the TFM to a level where it uses only one type of objects. After construction of problem domain object graph all the vertices with the same type of objects and operations must be merged, while keeping all relations with other graph vertices. As a result, object graph with direct links is defined. Schematic representation of class diagram development is given in Fig. 3.

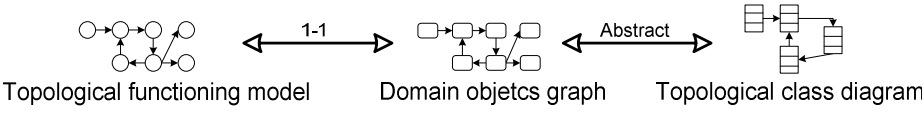


Fig. 3. Topological functioning model development process

In the [7] is offered development of UML class diagrams as the final step of the TFM usage within software system development process. In this class diagram relevant information – directions of associations between the classes – is lost. This important information is lost because within approach described in [7] the relations between classes are defined with one of the relations given in UML – the associations. It is not possible to transform topological (cause and effect) relations between TFM functional features into associations between classes. It is impossible because:

1. the direction of topological relation is not always the same as direction of association,
2. association also can be bidirected (topological relationship can not be bidirected), and
3. topological relationship only can be binary relation (association can relate more than two classes, for example, ternary association which relates three classes).

Because of this constraint in [7] it is recommended to define those association directions in further software development, for example, to develop a more detailed software design. But at this point a step back should be taken to review the TFM and its transformation on the conceptual class diagram. To avoid such regression and to save the obtained topology between the classes, by using the idea published in [5] about topological UML (TopUML) diagrams (including topological UML class diagrams), it is possible to develop a topological class diagram where the established TFM topology between classes is retained. In traditional software development relations (mostly associations and generalizations) between classes are defined by the modeler's discretion. The approach given in the [7] helps to identify associations between classes but the identification of direction for these associations again is defined by the modeler's discretion. By using within this research defined topological class diagrams and topological class diagrams development approach the relations between classes now are precisely defined according to constraints found between objects in problem domain.

4 Case Study of Using Topology within Software Development

For a better understanding of software development with the emphasis on topology (the construction of the TFM and development of the topological class diagram) let me consider small fragment of an informal description from the project, in which a document management system is developed.

4.1 Construction of Topological Functioning Model

Informal description of this project's problem domain is as follows: "When a document is received at the company, the secretary checks if this document has been already received. If the document is new, then secretary registers it in documents registry and gives to it new registration number. After that secretary gives archival number to it and puts it into the documents archive. If the new document demands an answer to it, then secretary delivers it to director. Director prepares answer document and gives it to secretary. Then secretary prepares copy of answer and sends the answer. Then secretary registers copy of answer in documents registry and performs all the actions which is performed when new document is received."

As given in 2nd chapter, the construction of TFM consists of three steps.

Step 1: Definition of physical or business functional characteristics.
For the document management system project example there are defined following ten functional features (in the form of tuple (as defined in [1]) containing the following parameters: identificator, object action (A), precondition (PrCond), object (O), mark if functional feature is external or internal), where "*In*" denotes Inner, and "*Ex*" – External:

<1, "Receiving a document", Ø, Document, Ex>, **<2**, "Checking the document", Ø, Secretary, In>, **<3**, "Registration of document", "If the document is new", Secretary, In >, **<4**, "Assigning registration number to document", Ø, Secretary, In >, **<5**, "Assigning archival number to document", Ø, Secretary, In >, **<6**, "Putting the document

in archive", Ø, Secretary, Ex>, <**7**, "Forwarding the document to director", "If the document demands answer on it", Secretary, In >, <**8**, "Preparation of answer document", Ø, Director, In >, <**9**, "Sending the answer", Ø, Secretary, Ex>, and <**10**, "Preparation of copy of answer document", Ø, Document, Ex>.

Step 2: Introduction of topology *Θ*.
Introduction of topology is done by introducing cause and effect relations (cause and effect relations are represented as arcs of a directed graph that are oriented from a cause vertex to an effect vertex) between functional features and thus defining topological space of a problem domain functioning (see Fig. 4).

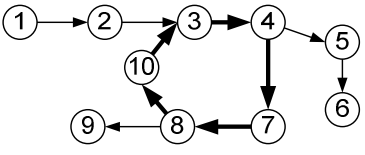


Fig. 4. Topological space of the document management functioning

In the Fig. 4 is clearly visible that cause and effect relations form functioning cycles. All cycles and sub-cycles should be carefully analyzed in order to completely identify existing functionality of the system. The main cycle (cycles) of system functioning (i.e., functionality that is vital for the system's life) must be found and analyzed before starting further analysis.

Step 3: Separation of the topological functioning model.
The example below illustrates how is performed the closuring operation over the set of inner functional features (*the set N*) in order to get all of the system's functionality – the TFM (*the set X*). The set of the system's inner functional features N = {2, 3, 4, 5, 7, 8, 10}. The set of external functional features M = {1, 6, 8}. The neighbourhood of each element of the set N is as follows: X_2 = {2, 3}, X_3 = {3, 4}, X_4 = {4, 5, 7}, X_5 = {5, 6}, X_7 = {7, 8}, X_8 = {8, 9, 10}, and X_{10} = {3, 10}.

The obtained set X (*the TFM*) = {2, 3, 4, 5, 6, 7, 8, 9, 10} (see Fig. 5).

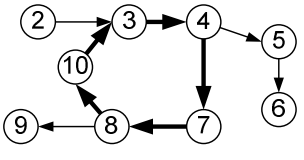


Fig. 5. Topological space of the document management functioning

The example represents the main functional cycle defined by the expert, which includes the following functional features "3-4-7-8-10-3" and is denoted by bold lines in Fig. 5. These functional features describe registering document and preparing answer on it.

4.2 Development of Topological Class Diagram

Before development of topological class diagrams it is needed to construct problem domain object graph (see Fig. 3). Problem domain object graph of document management is given in Fig. 6.

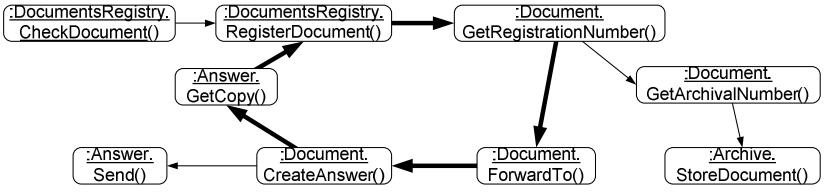


Fig. 6. The graph of problem domain objects with operations

After construction of problem domain objects graph it is possible to develop topological class diagram (see Fig. 3.). The developed topological class diagram of document management can be seen in Fig. 7.

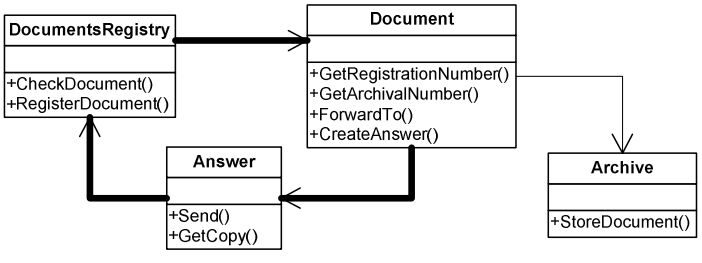


Fig. 7. The topological class diagram of document management system

With the boldest lines in developed topological class diagram is maintained main functional cycle which is defined by the expert within the constructed TFM. This reflects the idea proposed in [5] and [6] that the holistic domain representation by the means of the TFM enables identification of all necessary domain concepts and, even, enables to define their necessity for a successful implementation of the system.

5 Conclusions and Future Works

By using TFM within software development it is possible to introduce more formalism into UML diagrams (in this research formalization of UML diagrams is stated as formalization of information contained in UML diagrams) – it is possible to transform topology defined in TFM into UML diagrams. In this paper is described topology introduction into class diagrams and defined an approach for doing this introduction. It is possible to transform topology from TFM into class diagrams because TFM has strong mathematical basis. In this way the formalism of class diagrams means that between classes now are precisely defined relations which are identified from the

problem domain with help of TFM. In traditional software development scenario relations (mostly associations and generalizations) between classes are defined by the modeler's discretion.

While the methodology with three steps for construction of TFM is well defined, the method for creating sufficient informal description (the problem domain description) basing on which the TFM is constructed is not developed yet. When creating TFM of system's functioning a number of questions arise. The main question is: when the informal description of system functioning is sufficient for successful software system development?

Future research directions are as follows: study possibilities for introducing topology and in this way to increase formalization level of other UML diagrams, for example, activity diagrams; define method for creating sufficient informal description of problem domain for construction of TFM; and develop a tool that supports TFM and topological UML diagrams within software development process.

References

1. Asnina, E.: The Formal Approach to Problem Domain Modelling Within Model Driven Architecture. In: Proceedings of the 9th International Conference Information Systems Implementation and Modelling (ISIM 2006), Přerov, Czech Republic, pp. 97–104. Jan Štefan MARQ (2006)
2. Evans, A., Kent, S.: Core Meta-Modelling Semantics of UML: The pUML Approach. In: France, R.B., Rumpe, B. (eds.) UML 1999. LNCS, vol. 1723, pp. 140–155. Springer, Heidelberg (1999)
3. Fowler, M.: UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd edn. Addison-Wesley, Reading (2003)
4. Mellor, S., Balcer, M.: Executable UML: A Foundation for Model-Driven Architecture. Addison-Wesley, Reading (2002)
5. Osis, J.: Extension of Software Development Process for Mechatronic and Embedded Systems. In: Proceeding of the 32nd International Conference on Computer and Industrial Engineering, University of Limerick, Limerick, Ireland, pp. 305–310 (2003)
6. Osis, J.: Formal Computation Independent Model within the MDA Life Cycle. International Transactions on Systems Science and Applications 1(2), 159–166 (2006)
7. Osis, J., Asnina, E.: Enterprise Modeling for Information System Development within MDA. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), USA, p. 490 (2008)
8. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual, 2nd edn. Addison-Wesley, Reading (2004)
9. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language User Guide, 2nd edn. Addison-Wesley, Reading (2005)
10. Szlenk, M.: UML Static Models in Formal Approach. In: Meyer, B., Nawrocki, J.R., Walter, B. (eds.) CEE-SET 2007. LNCS, vol. 5082, pp. 129–142. Springer, Heidelberg (2008)
11. OMG: Unified Modeling Language Superstructure Specification, version 2.1.2 (2007)

Towards an XML Schema for Configuration of Project Management Information Systems: Conceptual Modelling

Solvita Bērziša

Faculty of Computer Science and Information Technology, Riga Technical University,
1 Kalku, Riga, LV-1658, Latvia
berzisa@gmail.com

Abstract. Project management is a complex process which is governed by project management methodologies, standards and other regulatory requirements. This process is supported by project management information systems, which could be configured according to requirements of particular methodologies represented in a standardized manner. This standardized representation is based on a comprehensive project management domain model. However, existing conceptualizations of the project management domain have limited scope. Therefore, this paper proposes to elaborate a comprehensive concept model of the project management domain to be used for development of XML schema for configuration of project management information systems. Development of the comprehensive concept model is based on analysis and integration of existing conceptualizations of project management and concept model integration. Four existing conceptualizations are integrated, and the integrated project management domain concept model incorporates data, process, and knowledge related entities.

Keywords: Conceptual modelling, schema integration, project management information system.

1 Introduction

Project management (PM) is a complex process involving planning, decision-making, execution and control activities. In order to ensure quality of PM processes, it is guided by various methodologies and standards that usually define PM processes, project structure, deliverables, templates and other items. There are general and domain specific PM methodologies. Project Management Body of Knowledge (PMBOK) [2] and PRINCE [3] belong to the general PM methodologies. The specific methodologies cover PM issues as well as product development issues. These methodologies are developed for certain domain area or individual organizations and funding agencies to meet their PM requirements. RUP [4] and MSF [5] are examples of specific PM methodologies in the software development area.

Regardless of PM methodology used, successful PM requires an appropriate project management information system (PMIS). It provides a framework to help guide the progress of project, because accurate, timely and relevant information is essential

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 229–237, 2010.

to the decision-making process of a project and inadequate information puts a project at risk. Large automated and integrated PMIS are usually developed on the basis of packaged PM applications. Ideally PMIS would adhere to all requirements of PM methodology used for a particular project. However, organizations have multiple projects often governed by different methodologies and regulatory requirements and implementation, and modification of PMIS becomes a complex task Therefore, an approach for implementation and configuration of PMIS was developed [1]. It uses principles of model and templates driven configuration of packaged applications to reduce efforts of implementing PMIS. An important part of this approach is a standardized definition of the project management domain [1]. PM methodologies and other project requirements are specified according to this standardized definition, which is subsequently used to configure a packaged project management application by means of package specific transformations. The standardized definition can be represented as a project management domain definition XML schema. This schema must be sufficiently comprehensive so that different PM methodologies can be described in terms of this definition.

In order to develop this schema referred as XML for Configuration of Project Management information systems (XCPM), conceptual modelling of the project management domain is performed. It is based on analysis and fusion of different existing definitions of the project management domain. Thus, an objective of this paper is to elaborate a comprehensive concept model of the PM domain to be used for development of XCPM. The concept model is a composition of different concepts relevant to PM. It is developed on the basis of existing conceptualizations of the PM domain, including XML schemas used in PM applications, PM ontologies and PM methodologies. A separate concept model is created for each of these sources. The concept model integration is performed to obtain the comprehensive concept model of the PM domain, which subsequently will be used to develop XCPM. The paper describes development of the separate concept models, integration process and the resulting PM concept model. The concept model is developed following an expansive approach to ensure that none of important concepts is left out. A more reductionist approach will be taken in development of XCPM.

The contribution of this research is development of the comprehensive definition of the PM domain suitable for further development of XCPM and PMIS configuration methods. The new model is developed because existing conceptualizations, which are analyzed in Section 2, have narrower scope and none of them contains all necessary items for configuration of PMIS. A distinctive feature of the developed concept model is incorporation of data, process and knowledge related entities.

The rest of the paper is structured as follows. Section 2 reviews literature about existent project management schemas, ontology and PM methodologies. Section 3 briefly describes the PMIS configuration process. Section 4 describes concepts model integration method and results. Section 5 concludes.

2 Literature Review

Existing PM XML schemas, PM ontologies and PM methodologies are used as sources for development of the comprehensive PM domain model.

2.1 Project Management XML Schemas

Two XML schemas for specification of PM data are available: PMXML [6, 7] and Microsoft Project XML schema [8]. These schemas are mainly intended for project data exchange between different PM software tools. Both definitions describe projects using four main concepts – project, task, resource and assignment.

PMXML [7] and MS Project XML [8] schema concept models are created based on the available schemes specifications. Fragment of the PMXML schema concept model and relationships are shown in Figure 1. Full information about this concept model and all others concept models referred in the article are available at http://www.berzisa.com/publication/ADBIS2009/Figures_ADBIS2009.html.

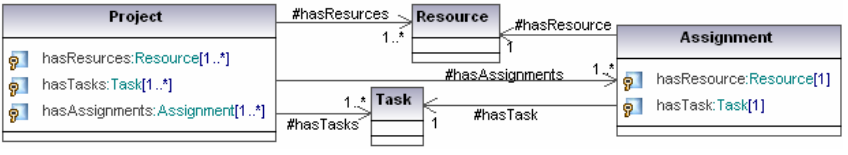


Fig. 1. Fragment of the PMXML concept model

2.2 Project Management Ontology

Several PM ontologies have been developed to define the PM domain. The most significant are PROMONT [9] and domain ontology for PM (PMO) [10]. PROMONT is proposed as reference ontology to support collaboration among multiple enterprises working on common projects [9]. PMO is a set of ontologies that captures and stores the PM knowledge [10].

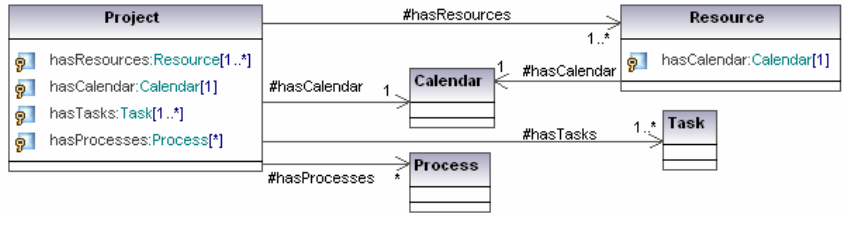


Fig. 2. Fragment of the PM ontology concept model

There are also some ontologies that describe the specific project elements, for example, project metrics ontology [11], ontology that describes the project team of a building project [12], project plan ontology for research program [13] and research project ontology [14].

All reviewed PM ontologies concepts and relations based on literature are joined together to develop a PM ontology concept model (fragment in Figure 2).

2.3 Project Management Methodologies

Information about the PM domain also can be found in methodologies and guidelines. Some of the best known generic PM methodologies are PMBOK and PRINCE and domain specific methodologies in software development are RUP and MSF.

PMBOK [2] describes general guidelines for PM that represents PM as closed loop of initializing, planning, executing, controlling and closing. Nine PM areas (e.g., integration, scope, cost, quality, risk management) are defined. Each area is described by presenting the management process and defining activities of the process by their input, tools and techniques and outputs. A similar methodology is PRINCE2 [3] that is a structured PM method based on best practice. Eight processes and eight components (areas of knowledge) are defined. Each process consists of the sub-processes which is used a certain input and output information. Only three techniques are defined, but also fourth technique is added – planning of activities and resources.

RUP [4] is one of the most complete software development methodologies developed by Rational Software (now part of IBM). It describes organization and execution of software development project. The methodology defines project roles. Each role performs a number of activities using specific tools and produces certain artefacts. Roles, activities and artefacts are describes in a standardized manner. For instance, each role is defined by description, required skills and activities and artefacts it is responsible for. It also defines general PM and software development workflows. The structure of MSF [5] is also very similar to RUP. This methodology defines roles, typical work items and work products and PM and product development workstreams.

The PM methodologies concept model is developed by using PMBOK as a basis and by adding concepts from other surveyed methodologies. This concept model consists of about 95 concepts and 218 relations (fragment in Figure 5.b.). The PMBOK is selected as the basis because it is generic and has wider reach than other methodologies considered.

The conceptualizations reviewed have limited scope and do not constitute as sufficient basis for developing XCPM. PMXML and MS Project XML define only operational PM data. PM ontologies focus on PM knowledge while data and process related aspects are represented only at very high level of abstraction. Knowledge and process are described in PM methodologies though this description is unstructured and, in the case of specific methodologies, represent just a single viewpoint of PM.

3 Overview of Configuration of PMIS

Development of XCPM is a part of ongoing research on development of the model and template driven approach for configuration of PMIS according to requirements of project management requirements and regulatory requirements. Figure 3 gives an overview of this approach (more detailed description can be found in [1]). The approach assumes that an organization aims to set up its PMIS according to requirements defined by a project management methodology or regulatory guidelines. These requirements are represented in a standardized manner. The standardized representation can be automatically transformed to configure a PMIS by using PM application specific transformation scripts. The standardized representation enables using the

same transformation scripts for different project management methodologies. XML based representation is chosen because of its compatibility with other technologies to be used in configuration of PMIS (e.g., process modelling and packaged applications) and wide usage in standardization.

The standardized representation is developed using XCPM. Therefore XCPM should be comprehensive enough to enable description of different PM methodologies. The analysis of the PM domain shows that XCPM should resemble data, process and knowledge related aspects of PM.

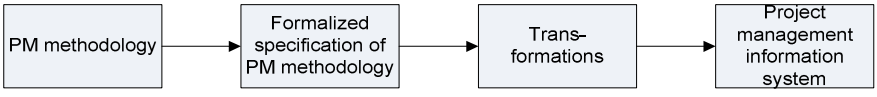


Fig. 3. An approach to configuration of project management systems

4 PM Concept Model Integration

Four concept models have been identified during the literature review. There are many common elements in these concept models, but at the same time each model has its specific concepts and attributes. In order to obtain a comprehensive PM domain concept model all four concept models are integrated using an algorithm described in Section 4.1. The integrated PM domain concept model is given in Section 4.2.

4.1 Model Integration Algorithm

Although majority of data model integration methods are developed for integration of database models or XML schemas, the same principles and techniques can be used for integration of concept models. The model integration consists of the following main tasks: pre-integration, model (schema) conforming, schema merging and schema restructuring [15, 16]. The concept models are created in pre-integration phase. During the conforming phase conflict detection and conflict resolution are performed. Three common types of the conflicts are naming, semantic, and structural conflicts [17]. Three main types of schema conforming transformations are entity/attribute, entity/relationship, and attribute/subtype equivalence [15]. Related concepts are identified and a single federated concept model is produced in the merging phase. Four types or schema merging transformations are used, namely, entity merge, introduction of is-a relationships, addition of union, and addition of intersection [15]. During schema restructuring quality of the federated concept model is improved [15, 16]. Four types of schema restructuring transformations are used – redundant attribute removal, optional attribute removal, generalization of attributes, and redundant relationship removal [15]. Schema conforming and schema merging are executed in an iterative manner in each stage adding one concept model.

In order to establish the integrated PM domain concept model, all four source models are integrated step-by-step starting with integration of PMXML and MS Project XML concept models by applying aforementioned model integration mechanisms (Figure 4). Nine steps of the integration algorithm are: 1) the pre-integration step,

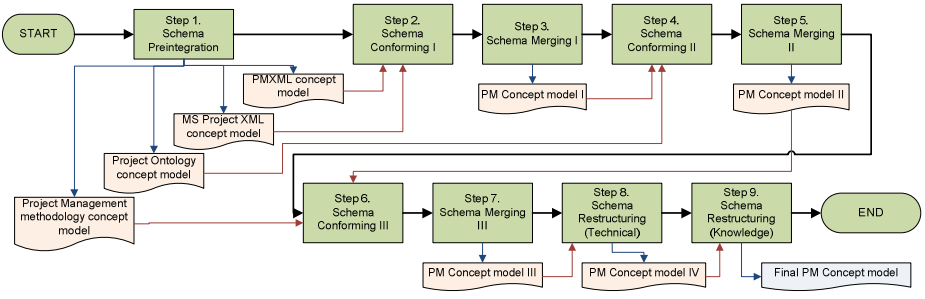


Fig. 4. PM concept model integration algorithm

which was performed during the literature review, and four PM concept models are produced as the result; 2) the PMXML and Microsoft Project XML concept models conforming step; 3) the PMXML and Microsoft Project XML concept models merging step, which yields a federated PM concept model I; 4) the PM concept model I and the PM ontology concept model conforming step; 5) the PM concept model I and PM ontology concept model merging step, which yields a federated PM concept model II; 6) the PM concept model II and PM methodology concept model conforming step; 7) the PM concept model II and PM methodology concept model merging step, which yields a federated PM concept model III; 8) the PM concept model III technical restructuring step, which yields a PM concept model IV; 9) the PM concept model IV knowledge based restructuring step, which yields the final PM concept model. All concept models generated during the integration process can be viewed in web page.

Schema conforming and merging transformations are used to perform individual steps of the process. Details of application of these transformations are presented for Step 6 and Step 7 using simplified examples of concept models (Figure 5). Input data of the schema conforming activity in Step 6 are simple examples of PM concept model II (Figure 5.a. – obtained by integrating the concept models of Figure 1 and Figure 2) and the PM methodology concept model (Figure 5.b). One synonym conflict is detected in this example. Entity *Task* in PM concept model II means the same as *Activity* in PM management concept model. To eliminate this synonym conflict, entity *Task* is renamed to *Activity*. According to the entity/relationship equivalence, entities *Project* and *Process* are directly related in PM concept model II, but in the PM methodology model through entity *Knowledge Areas*. To solve this conflict, a new entity *Knowledge Areas* is created in PM concept model II that links *Project* with *Process.* Outputs of the Step 6 schema conforming activity are modified PM concept model II and the PM methodology concept model. Entity/attribute and attribute/subtype equivalences have not been used in this model conforming example. In Step 7, schema merging uses these output models and generates PM concept model III. Initially, schema merging unites entities that are in both models using the entity merge transformation. Then entities, which are only in one of the models, are added to the new model using the addition of units/intersection transformation. In this example, a new entity from PM concept model II is *Calendar*, but a new entity from PM methodology concept model is *Risk*. The result of this conforming and merging example is shown in Figure 6.

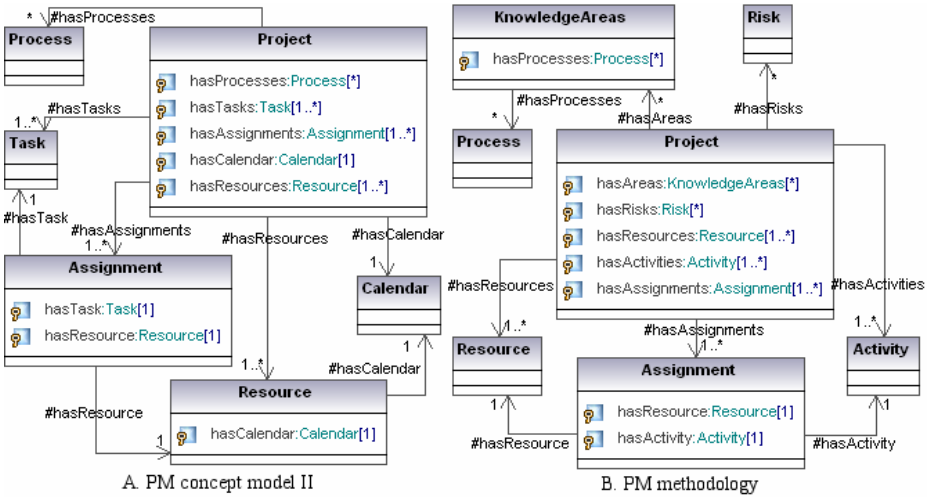


Fig. 5. Concept models fragments before integration

4.2 PM Concept Model

After completing the integration process, which was described in Section 4.1 the PM concept model is obtained. This model consists of 104 entities and has 243 relationships. A fragment of the model is shown in Figure 6. The model defines all main entities and relationships present in the project management domain approached from the information systems perspective.

The central entity of the model is *Project*. This element is directly or transitionary related to sets of entities describing project planning and project controlling. The *KnowledgeArea* entity defines well-established PM knowledge areas describing knowledge, skills and processes needed for successful PM. It represents a generic case, while there are several entities representing specific knowledge related entities such as *RiskManagement* and *IntegrationManagement.* The knowledge related entities are associated with the *Project* entity through other entities, for instance, *Schedule-Management* is associated with *Project* through the *Activity* entity. The *Process* entity

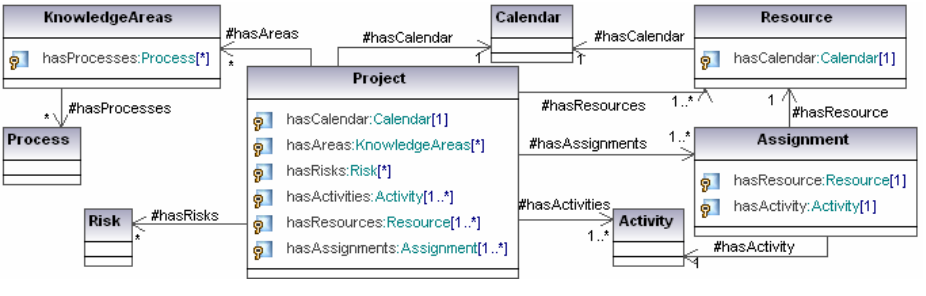


Fig. 6. Fragment of PM concept model

represents various PM processes, which are performed as part of quality management, risk management, issues management and other PM activities.

The concept model is comprehensive enough to be suitable for representing different project management methodologies, regulatory requirements and guidelines. That is implied by diminishing number of model modifications made at each consecutive step of the concept model integration process.

5 Conclusions

The paper described development process of the PM domain concept model. PM concept model development process consists of two parts: the literature review and the concept model integration process. During the literature review, four separate concept models are created from existing conceptualizations, including PM software XML schemas, PM ontologies and descriptions of PM methodologies. During the integration process, existing models are integrated together by applying schema conforming and merging transformations in the nine steps process. The result of the integrating process is the comprehensive PM concept model that in future will be used for development of XML schema for configuration of PMIS. The distinctive feature of the model is representation of data, process and knowledge related entities, what is important for configuration of PMIS. The paper and future research on configuration of PMIS also contributes to the area of automated configuration of packaged applications. The PM domain is particularly well-suited for automated configuration of packaged applications because it is relatively well-defined domain and has high level of uniformity from case to case.

This article is the first part of XCPM schema design that describes main schema elements and defines its theoretical basis and justification. Definition of XCPM schema structure will be the next step of future research. The concept model has relatively complex structure, which will be optimized and streamlined during the schema development. Some entities, relationships and attributes may be changed in relation to the developed PM concept model. The main steps of future research are elaboration and approbation of XCPM schema and it transformation to PMIS.

Acknowledgment

The author would like to acknowledge Dr.sc.ing. Jānis Grabis for his scientific guidance and support and reviewers for their valuable comments.

References

1. Bērziša, S., Grabis, J.: An Approach for Implementing of Project Management Information Systems. In: Information Systems Development: Towards of Services Provision Society. Springer, New York (2008)
2. Project Management Institute: Guide to the Project Management Body of Knowledge (PMBOK), Project Management Institute, Newton Square (2004)

3. Hedeman, B., Heemst, G.V., van Fredriksz, H.: Project Management Based on PRINCE2 – PRINCE2 Edition 2005. Van Haren Publishing (2007)
4. Kroll, P., Kruchten, P.: The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP. Addison-Wesley, New York (2005)
5. Turner, M.S.V.: Microsoft Solutions Framework Essentials: Building Successful Technology Solutions. Microsoft Press, Redmond (2006)
6. Cover Pages, Project Management XML Schema (PMXML), http://xml.coverpages.org/projectManageSchema.html
7. Project Management XML Schema (PMXML), http://schemas.liquid-technologies.co/Other/PMXML/2.0/
8. XML Structure for Microsoft Office Project 2003, http://msdn.microsoft.com/en-us/library/aa210619office.11.aspx
9. Abels, S., Ahlemann, F., Hahn, A., Hausmann, K., Strickmann, J.: PROMONT - A Project Management Ontology as a Reference for Virtual Project Organizations. In: On the Move to Meaningful Internet Systems 2006, OTM 2006 Workshops, pp. 813–823 (2006)
10. Ríz-Bertol, F.J., Dolado, J.: A Domain Ontology for Project Management. In: Berki, E., Nummenma, J., Sunley, I., Ross, M., Staples, G. (eds.) Software Quality Management XV: Software Quality in the Knowledge Society, vol. 15, pp. 317–326 (2007)
11. Project Metrics Ontology, http://www.daml.org/2002/03/metrics/metrics-ont
12. RDF schema to describe the project team of a building project, http://www.daml.org/ontologies/306
13. Summary project plan ontology targeted to a research program, http://www.daml.org/2001/02/projectplan/projectplan
14. Research project ontology, http://www.daml.org/ontologies/3
15. McBrien, P., Poulovassilis, A.: A formalisation of semantic schema integration. Information Systems 23(5), 307–334 (1998)
16. Batini, C., Lenzerini, M.: A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 18(4), 323–364 (1986)
17. Aleksandraviciene, A., Butleris, R.: A Comparative Review of Approaches for Database Schema Integration. In: Advances in Information Systems Development, pp. 111–122. Springer, US (2007)

Clustering XML Documents by Structure

Anna Lesniewska

Institute of Computing Science, Poznan University of Technology,
Piotrowo 2, 60-965 Poznan, Poland
alesniewska@cs.put.poznan.pl

Abstract. Clustering of XML documents is an important data mining method, the aim of which is the grouping of similar XML documents. The issue of clustering XML documents by structure is being considered in this paper. Two different and independent methods of clustering XML documents by structure are being proposed. The first method represents a set of XML documents as a set of labels. The second method introduces a new representation of a set of XML documents, which is called the SuperTree. In this paper, it is suggested that the proposed methods may improve the accuracy of XML clustering by structure. Such thesis is based on the tests, the aim of which is to assess advantages of the proposals, as conducted respectively on the heterogeneous and homogenous sets of data.

Keywords: XML, data mining, similarity measure.

1 Introduction

XML is becoming a standard for developing numerous web applications that deal with document retrieval and storage. There is a number of articles that have raised the issue of processing, storing and management of XML documents. However, in the field of XML processing, mining of XML documents has become a new topic. Clustering of XML documents definitively represents one of the most interesting trends in this research area. The process aims at grouping together similar XML documents, though the problem of data clustering has been considered in various context throughout many years. This has led to the development of several methods. At this point, it is worth mentioning that XML documents differ from traditional data. An XML document consists of a structure and content. If follows therefore that the existing clustering algorithms are considered to be inappropriate for the purposes of clustering XML documents [1]. Thus, there is a need to develop a new clustering algorithm specifically for XML documents. Those can be categorised in separate groups. The first group of methods is based on both, structure and the content of XML documents, while the second relies solely on the structure.

The clustering of XML documents by structure has many applications. Identification of XML documents with similar structure can prove useful while focusing on detection of structural similarities among documents. Consequently, this helps to solve the problem of recognition of different sources that provide the same kind of information, or alternatively in the structural analysis of a Web site. The documents

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 238–246, 2010.

that represent the same information may be presented differently, however the structure of the documents can be similar. XML documents may be presented in many ways. They can be shown as graphs, trees, set of paths, set of labels, time series etc. The representation of XML documents has a crucial impact on both, the quality and efficiency of the clustering process. Another important factor determining the quality and efficiency of this procedure with regard to XML documents is the character of the source of data. The distinction shall be made between the heterogeneous and homogenous sources. Processing the data from the heterogeneous source is easier than from a homogenous one as the documents from the heterogeneous set of data have different structure features (labels, paths etc.), which are generally easier to find. In the case of homogenous data the method must encompass the structure features instead of the labels, therefore the method should be considered as more complex.

The aim of this article is to present two various methods, which can be used in the process of clustering XML documents by structure. The first method is based on labels from which the vectors representing a given document are construed. The second one relies on paths to generate patterns, on the basis of which the clusters are created. In this method a new structure is being proposed (SuperTree), which represents a set of documents. To illustrate, SuperTree can be considered as the main tree construed from the set of paths, which consists all of the documents in the form of subtrees. The proposed methods are tested as against the heterogenous and homogenous sets of data. The preliminary results of the research have been illustrated in this paper.

An adequate and specific analysis is essential for the suggested method of clustering XML documents by structure and can be illustrated as follows: the choice of representation of XML documents, selection of appropriate measures for each representation, selection of appropriate algorithms for each pair of representation and measures, experimental evaluation.

The structure of the paper can be presented as follows: Section 2 presents relevant work on the issue of clustering XML documents by structure; Section 3 contains the problem formulation; Section 4 presents the basic notions; Section 5 describes proposed solutions; Section 6 shows the achieved results; Section 7 consists of conclusion and the future direction.

2 Related Works

So far the problem of clustering XML documents by structure has not received a great deal of attention, therefore there are only few solutions proposed. In the literature there are various areas of research relating to either representation of XML documents or similarity measures and clustering by structure. In order to come up with a proposition of an end-to-end method of clustering XML documents by structure it is necessary to include each of the research areas. The layout of the document can vary and could be shown as respectively a tree, a graph [10], sets of paths [3][4], time series [6], vectors and others like the BitCube proposed in [7] or SOM (Self-Organizing Map) in [8]. Most of the existing models for the representation of XML documents are based on labelled tree [5][9], as it is a natural illustration indicating a hierarchical structure of XML documents. An XML document transforms into a rooted labelled tree. Every element from the document stands for a node in the tree. Every node has

its own name and is represented by a label. An edge portrays an existing relationship between the nodes. The method of computing similarities between structures of XML documents varies accordingly. If the solution is based on the tree, the authors have used tree edit distance to measure the similarity between the structures of documents. If the demonstration is based on the graph, the similarity is being calculated on the basis of the set of nodes and edges [10]. In [11] the authors have computed structural similarity against the source of XML Schemas to domain the XML Schema. In [12] the authors have proposed solution based on the path. They have used sequential pattern mining to extract the frequent paths from XML documents and then used them for clustering.

The similarity measures may constitute an appropriate means for choosing representation. The survey about similarity measures for XML documents may be found in [14].

The sole method of grouping, namely algorithm, is a consequence of both, a chosen representation and the similarity measures. At this stage it is possible to separate two different paths. Firstly, an adaptation of existing grouping methods as taken from, the classic approach and secondly, a proposal of a new algorithm. In the solutions suggested by this paper more emphasis has been put on representation and an adequate manner of measuring similarities of documents. A grouping algorithm is rather a consequence of the two preceding steps.

3 Problem Formulation

The problem of clustering XML documents by structure can be stated as follows. Given an input set of XML documents (for example web documents) as well as the representation of these documents and a similarity measure as defined for a specific representation, the aim of XML documents clustering is to group together structurally similar documents. From the above statement it is possible to appoint the factors. Both the representation of XML documents and the similarity measures accepted for the representation play a crucial role in mining XML documents. As previously mentioned, the representation of XML documents may vary. The XML documents may be represented as a tree, as a graph, as a time series, as a vector or as the set of paths. Similarity measure as applied in clustering algorithm must be appropriate to a specific representation of the XML documents. In the case of a path, the similarity measure is usually based on distance metrics (Euclidean, norm L_1, L_2, etc.).

The next section presents an XML document representation and the similarity measure accordingly to the representation applied in our methods.

4 Basic Notions

Two XML document representations, which model the XML document representation as set of labels or set of paths will now be presented. Afterwards the similarity measures, which are going to be used accordingly in our experiments will be introduced.

4.1 XML Document Representation

To illustrate, assume that there is a set of XML documents D. Each document D_i is represented as a labelled tree T in which labels are denoted the nodes of the tree. A *path* is a sequence of nodes from the root to node in a tree T. The *absolute path* is a path from the root to the node (leaf) in the tree T. Given that the set of labels which is denoted as $L=(l_1,\ldots,l_m)$ is a set of distinct labels from documents, enumerate labels from L with successive numbers start with 1 to m ($|L|$=m, $\in N$>0), where m is a positive integer. The document $D_i \in D$ is represented by a *vector* of coefficient: $v(D_i)=<u_{i1},u_{i2},\ldots u_{im}>$, where each u_{ik} denotes the number of occurrences of a label l_i in the document D_i.

The document $D_i \in D$ is represented by a set of absolute paths: $P(D_i)=<p_{i1},p_{i2},\ldots,p_{ik}>$, where p_{ij} is the absolute path in the document D_i.

4.2 Similarity Measure

The similarity of a pair of documents D_i and D_j is defined with regard to the set of labels as follows:

$$sim_1(D_i,D_j)=\cos(v(D_i),v(D_j))=\frac{v(D_i)\circ v(D_j)}{\|v(D_i)\|\|v(D_j)\|}=\frac{\sum_{k=1}^{m}u_{ik}*u_{jk}}{\sqrt{\sum_{k=1}^{m}u_{ik}^{\,2}}*\sqrt{\sum_{k=1}^{m}u_{jk}^{\,2}}} \quad (1)$$

The similarity between documents $sim_1(D_i,D_j)$ depends on the angle between their vector representations, which is achieved by computing cosine coefficient for the vectors. Since every u_{ik} takes only non-negative values, $sim_1(D_i,D_j) \in <0,1>$. For instance u_{ik} can be defined as follows:

$$u_{ik}=\begin{cases}1, & \textit{label k exists in document i}\\ 0, & \textit{otherwise}\end{cases} \quad (2)$$

The similarity between a pair of documents D_i and D_j may be defined with regard to a set of paths as follows:

$$sim_2(D_i,D_j)=\frac{P(D_i)\cap P(D_j)}{P(D_i)\cup P(D_j)} \quad (3)$$

$P(D_i)$,$P(D_j)$ denotes set of absolute path in document D_i,D_j respectively.

5 Proposed Solution

In this section we illustrate two methods of clustering XML documents by structure based on the presented representation of XML documents. The first method is based

on representation of a set of XML data as a set of labels. The second one is relying on the representation of a set of XML data as a set of paths.

The first method is aimed at small, heterogenous collection of XML documents. It is not applicable though to a bigger and homogenous set of data. This is because for that kind of data the method proved not to be effective. The second method, however, may be applied to both types of data. In *Figure 1* a schema of the proposed approach based on labels is presented and in the *Figure 2* a schema of an approach based on paths is being illustrated. In the input there is a set of XML documents, and in the output clusters of structurally similar documents are achieved (denote C).

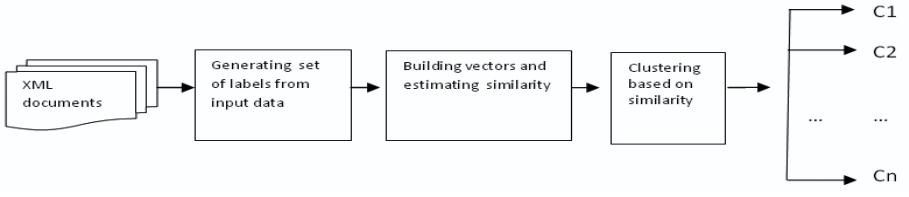


Fig. 1. Schema of the approach based on labels

5.1 Representation Based on Labels

Preparation - the input set of XML documents is transformed into a set of labels. The set of labels is transformed to a vector for further processing.

Structural similarity - prepared vector is used to estimate the structural similarity between documents. To compute structural similarity the cosine measure presented in Section 4.2 is used. The comparison is made between all documents as well as between one another and themselves respectively.

Clustering - in order to cluster the data, an algorithm AHC (Agglomerative Hierarchical Algorithm) is used to end up with the structurally similar groups. In the implementation, the algorithm does not calculate similarity between documents but is given this similarity in advance. Levels identifier corresponds to steps in the algorithm. Input data should be recorded in the database tables. The next thing is running adequate PL/SQL procedure and reading output data from specified database tables.

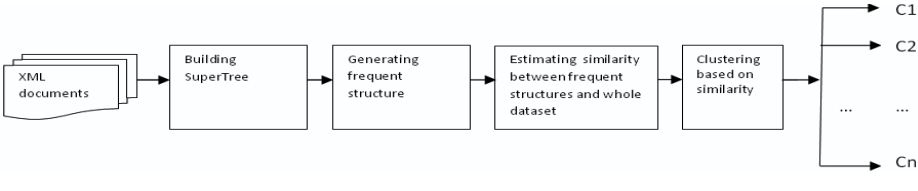


Fig. 2. Schema of the approach based on paths

5.2 Representation Based on Paths

The input set of XML documents is transformed into new representation called the SuperTree. The SuperTree is a representation based on labelled tree (set of paths).

This new concept could be defined as the tree, which will include all the documents (every document is a subtree in the SuperTree). From the SuperTree the patterns of structures are being generated, which form the basis for further processing (pattern of cluster).

Preparation - the process in this method is more complex. Initially, the patterns of structure must be derived. Then, basing on these patterns of structure the structural similarity of documents will be compared.

Summarize the document – in order to reduce the size of the SuperTree, the documents were summarised. The absolute path has been reduced until the unique appearance of a given path in a tree has emerged. Due to this, the size and the structure of the SuperTree have been considerably reduced.

Building a SuperTree – an inspiration for building the SuperTree was the FP-Tree from algorithm FP-Growth [13] used to find frequent patterns. Every XML document can be represented as a labelled tree, where nodes correspond to the tags of an XML document. Two nodes are connected if there is a relationship between them. The SuperTree is a minimal tree that contains every tree from its input. In other words, every single tree from the input is a subtree of the SuperTree. To build the SuperTree, each input tree in the collection is recursively traversed. SuperTree links all documents by a common added root called the 'root' (root label). Given the tree, each node is searched in the SuperTree. If this node does not exist, it is inserted. At the end of the process of constructing the SuperTree, the SuperTrees nodes are labelled with integer numbers in breadth first search order. For example we present five structures of XML documents, as presented on the *Figure 3* and the build of the SuperTree as presented in the *Figure 4.*

Fig. 3. Structure of exemplary documents

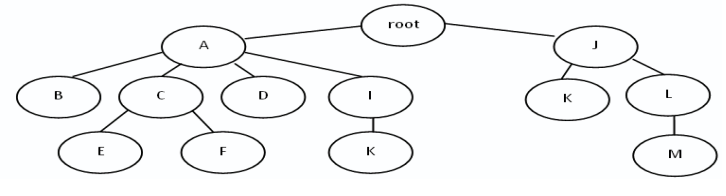


Fig. 4. The SuperTree

Generate patterns – based on the SuperTree structure frequent patterns are searched for. Patterns are being construed from the paths, which satisfy the minimal support (the number of occurrence of the paths in the whole dataset, given as input parameter)

Clustering – the already obtained frequent patterns are compared with the input data set using similarity measures described in section 4.2. In this way the output clusters with structurally similar XML documents that generated the basis for frequent patterns have been achieved.

6 Experiments

In this section the preliminary results and the plan of further research proposed are being presented.

First of all, the research was conducted on the basis of heterogenous data set including 1120 documents from three different sources (articles from Wikipedia [2], IMDB and RSS). The main goal of these experiments was to achieve clusters representing their original sources. The second sets of data were homogenous the MovieDB [2]. The MovieDB is a collection of XML documents describing movies. It contains 4820 XML documents. There are eleven defined structural categories, which correspond to the transformations of the original data structure. The experiment based on the application of the first method for the heterogenous set of data gave very good results, which are presented in *Tables 1*. Clusters involving aboriginal category of documents (source of documents) have been received. Mdb category appears twice, because this dataset possesses two sorts of labels, namely an original label like an 'author', 'title' etc. and the mapped one like 'a','b' etc. The proposed solution took these differences under consideration.

Table 1. Results for heterogenous set of data

Aboriginal category	Cluster	Docs
Mdb	1	393
Inex	2	500
Mdb	3	120
Rss	4	107

The second method for the same data gave as good results as the first one. Additionally, second method was tested for homogenous data. The XML documents from MovieDB, which were mentioned above, have created the structure of the SuperTree consisting of 433 nodes. Six common structures have been obtained (patterns of clusters), which formed the basis for further processing. It is also known that the cluster is build from eleven structural groups, so it is expected to achieve the same number of patterns at this stage. The proposed solution, however, was based on an assumption that the patterns were disjoined. The results of the experiment are presented in *Table 2*. It is possible to conclude that the information about further structures is contained in six structures, and therefore the similarity in the others is so low. This situation is bringing on repetition of absolute path in different cluster, which contain this

Table 2. Experiments on MovieDB Collection

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
C1	598	0	0	0	0	0	0	0	0	0	0
C2	0	486	0	0	0	0	0	0	0	0	0
C3	0	690	0	0	0	0	0	0	0	0	0
C4	0	0	0	172	0	0	0	0	0	0	0
C5	0	0	0	0	435	0	0	0	0	0	0
C6	0	0	0	0	0	231	0	0	0	0	0
C7	0	0	0	0	0	0	261	0	0	0	0
C8	0	0	0	0	0	769	0	0	0	0	0
C9	0	0	0	0	0	0	172	0	0	0	0
C10	0	0	0	0	0	387	0	0	0	0	0
C11	0	0	0	0	0	448	0	0	0	0	0
Purity	**1**	**0.41**	**0**	**1**	**1**	**0.12**	**0.6**	**0**	**0**	**0**	**0**

set of data. The data in the *Table 2 C* as presented in the row indicate an expected class of documents. Contrastingly, the data under C in the column denote an obtained cluster of documents. As an evaluation measure the 'Purity' measure had been adopted, which assesses the relationship between the accuracy adjustment and the class of documents.

This particular dataset has been utilised because the information about the class, to which the document belongs, have been known. At this point, the main aim was to obtain the most precise and the 'purest' possible set of documents belonging to one class. The measure of 'Purity' indicates the accuracy with which the document has been adapted to a specific class. Had the group of documents contained the unique pattern our proposed solution was adequate (as was the case with data in C1, C4 and C5). In the other set of documents belonging to one class there were documents matching the isolated pattern. However, in that instance, one class of documents contained the paths, which were repeatable in the other classes, in which case the proposed solution proved to be ineffective. As a consequence, a group of documents belonging to one class have turned out to be empty. This means that a given class of documents has not been defined by the pattern, on which our method is based. The problem of common pattern will be considered in the further research.

7 Conclusions

In this paper two different approaches of clustering documents by structure are proposed. The first method focuses on the heterogonous and small collection of data. The second method may be applied to both types of sources of data (heterogonous and homogonous) as well as to the larger data.

The research shows that if there is a group of documents from different sources and the purpose is to assign documents to a given category (source), the labels are sufficient to meet this goal. In the case of homogenous sources the problem is more complicated and requires a more refined approach. If the homogenous sources are dealt with, it is not possible to rely solely on the labels as the whole cluster will have the

same ones. Therefore another approach is preferable. The SuperTree representation satisfied the 'missing' requirements, however further research involving large sets of data is needed in order to confirm the results.

At this stage, the evaluation method to improve quality of clusters, have not been used. To estimate the accuracy of the results the 'Purity' measure was utilised. In the future, however, other possibilities will be considered. In the above research the original label of nodes was used, without distinction on the upper or lower case letter or semantic meanings (synonyms, homonyms etc.) There are plans, thus, to use embedded subtrees in spite of the original tree. Further research on structural similarity, using various test data, is being planned in the future. There are also plans to conduct the research focused on finding the dependence between the structure and content of the documents and to answer the question - whether we can define the content (categories) of the document based solely on the structure.

References

1. Dalamagas, T., Cheng, T., Winkiel, K., Sellis, T.: Clustering XML documents by structure. In: EDBT Workshop 2004, pp. 547–556 (2004)
2. Wikipedia XML Corpus (2006), http://www-connex.lip6.fr/~denoyer/wikipediaXML
3. Vercoustre, A., Fegas, M., Gul, S., Lechevallier, Y.: A Flexible Structured-based Representation for XML Document Mining, inria-00000839 (2006)
4. Rafiei, D., Moise, D.L., Sun, D.: Finding Syntactic Similarities Between XML Documents. In: Proceedings of the 17th International Conference on Database and Expert Systems Applications (2006)
5. Candillier, L., Tellier, I., Torre, F.: Transforming XML trees for efficient classification and clustering. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 469–480. Springer, Heidelberg (2006)
6. Flesca, S., Manco, G., Masciari, E., Pontieri, L.: Fast Detection of XML Structural Similarity. IEEE Transactions on Knowledge and Data Engineering (2005)
7. Yoon, J.P., Raghavan, V., Chakilam, V.: BitCube: A three-dimensional bitmap indexing for XML documents. In: SSDBM, Fairfax,Virginia, USA, pp. 158–167 (2001)
8. Hagenbuchner, M., Trentini, F., Sperduti, A., Scarselli, F., Tsoi, A.C.: A Self - Organising Map Approach for Clustering of XML Documents, pp. 1805–1812. IEEE, Los Alamitos (2006)
9. Nayak, R., Iryadi, W.: XMine: A Methodology for Mining XML Structure. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 786–792. Springer, Heidelberg (2006)
10. Lian, W., Cheung, D.W., Mamoulis, N., You, S.: An efficient and scalable algorithm for clustering XML documents by structure. IEEE Trans. Knowl. Data Eng. (2004)
11. Li, J., Liu, C., Yu, J., Liu, J., Wang, G., Yang, C.: Computing Structural Similarity of Source XML Schemas against Domain XML Schema. In: Proc. 19th Australasian Database Conference, Wollongong, Australia (2008)
12. Garboni, C., Masseglia, F., Trousse, B.: Sequential Pattern Mining for Structure-Based XML Document Classification. In: Advances in XML Information Retrieval and Evaluation, pp. 458–468. Springer, Heidelberg (2006)
13. Han, J., Pei, J.: Mining Frequent Patterns by Pattern-growth: Methodology and Implications. ACM SIGKDD Explorations (2000)
14. Tekli, J., Chbeir, R., Yetongton, K.: An overview on XML similarity: Background, current trends and future directions. Computer Science Review 3(3), 151–173 (2009)

Multidimensional Business Process Modeling Approach

Ligita Businska

Department of Systems Theory and Design, Riga Technical University,
1 Kalku, Riga, LV-1658, Latvia
Ligita.Businska@cs.rtu.lv

Abstract. The paper discusses an approach that enables incremental business process modeling and analysis from multitude set of perspectives, thus combining wide spectrum of business process characteristics in a common multidimensional model, and allowing the translation of business process modeled in a particular perspective into other perspectives. The paper presents basic terms and general stages for iterative construction of a multidimensional business process model.

Keywords: business process, multidimensional modeling, business process model, modeling perspective, modeling dimension.

1 Introduction

Organizations have many reasons to capture their business processes (BPs). Companies, which have merged, may want to examine processes across their lines of business to discover which one is the best of breed; or companies may need to improve their existing processes, or to automate them by developing appropriate information systems [1]. Usually BP models are constructed to create a knowledge base that could satisfy different purposes, i.e., BP management, reengineering, integration, monitoring, etc. The set of modeling dimensions of BP included in the model depends on particular modeling goal and the available and desired knowledge about this BP. In this paper we regard dimension as a particular BP parameter or characteristic, where the group of dimensions represents the part of n-dimensional business process model, considered from a certain perspective. In BP modeling it is usually not possible to consider many dimensions at one go. Instead, the model is constructed gradually, according to growing knowledge about BP obtained by different ways of its analysis, as a result of a changing focus of modeling goals, adding new modeling dimensions, switching to a different set of dimensions and reusing already constructed BP models.

Thus, on the one hand, there is the need for an approach that combines a wide spectrum of BP characteristics in a common multidimensional model and allows to construct it incrementally with possibility to translate the BP model built from a particular perspective to other perspectives. On the other hand, literature offers a wide range of frameworks (for instance, enterprise modeling environments) that provide the opportunity to capture knowledge about BP and examine it from multiple viewpoints. However, in most cases it is necessary to develop each model separately and to apply distinct modeling techniques, because usually the methods and techniques for BP

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 247–256, 2010.

modeling and analysis express only some aspects of the processes (e.g. activities, roles, interactions, data, etc.) and are applicable in limited scope of application areas [2]. Consequently, it is necessary to combine several modeling methods and techniques to achieve the desired results. Limited possibilities of reuse of BP modeling artifacts and necessity to maintain several modeling environments lead to non-effective use of enterprise time and recourses. To accommodate the diversity of process modeling and analysis knowledge we need a unified multidimensional modeling space, which could provide an opportunity not only to view, but also to construct incrementally and analyze BP in a dimension sensitive level of conceptual granularity, corresponding to evolving business knowledge and multiple modeling purposes [3].

The paper presents initial results of doctoral work discussing general issues of the proposed multidimensional BP modeling approach. This investigation is a branch of broader research of multidimensional BP model (MBPM) development carried out at the Riga Technical University. The purpose of the doctoral research is to develop a problem domain oriented BP modeling approach that allows to analyze BP parameters according to the set of dimensions that is appropriate for a particular business situation. The discussion in this paper concerns only basic issues of multidimensional BP modeling, such as the basic issues of multidimensional BP modeling and the conceptual guidelines of how to construct MBPM.

The paper is structured as follows – Section 2 briefly presents related works relevant to the research problem. Section 3 describes the research method, possible research stages and deliverables. Section 4 gives an overview of initial results thus presenting basic concepts of a MBPM and guidelines for multidimensional BP modeling. In Section 5 conclusions outline some directions of the future work and current and expected final contribution of the research work.

2 Related Work

Over the years the scope of BPs and BP modeling has broadened. Less than a decade ago, BP modeling was known as a method for managing and driving largely human-based, paper-driven processes. Nowadays BP modeling is an enterprise integration technology applicable for a wide spectrum of objectives, complementing such specific tasks as Service-Oriented Architecture (SOA), Enterprise Application Integration (EAI), and Enterprise Service Bus (ESB) [4]. Many forms of information must be integrated into a process model in order to satisfy possible BP modeling tasks, including perspectives that just cover the detailed sequence of the process and eventually adding perspectives that address specific BP context characteristics such as business goals, performance measures, deliverables, process owner, process type, customer and other process characteristics presented by essential process theory. In most cases BP modeling languages and techniques (e.g., UML2 Activity Diagram, BPMN, Event Driven Process Chain, IDEF3, Petri Net, Role Activity Diagram) permit the use of functional and behavioral decomposition of the process, while the organizational and informational perspectives are only partly supported [5]. However, these modeling languages and techniques do not support BP context fully. Consequently, if it is necessary to analyze a wide spectrum of BP characteristics, a multiple set of modeling techniques and languages should be applied. Besides, tools that support BP modeling

do not facilitate a unified modeling space. Several tools such as GRADE [7], ARIS [8], JOpera [9], EML [10], IBM WebSphere [11] and BPMO [12] in most cases provide modeling by just some dimensions dealing with one and the same process model and make emphasis on the representation facilities ensuring the choice of personalized view [13]. Thus, we could conclude that in most cases it is not possible: a) to model a BP incrementally from different perspectives simultaneously; b) to construct several process models that reflect different views on the same BP with possibility to switch between models with different set of BP characteristics; 3) to translate BP modeled in a particular set of dimensions to other set of dimensions. Though there are transformation patterns and wizards that are developed for BPs, they are used for simulation and optimization tasks [6] or for the process model transformation into executable code [5]. The present situation in the areas relevant to BP modeling is reflected in Figure 1.

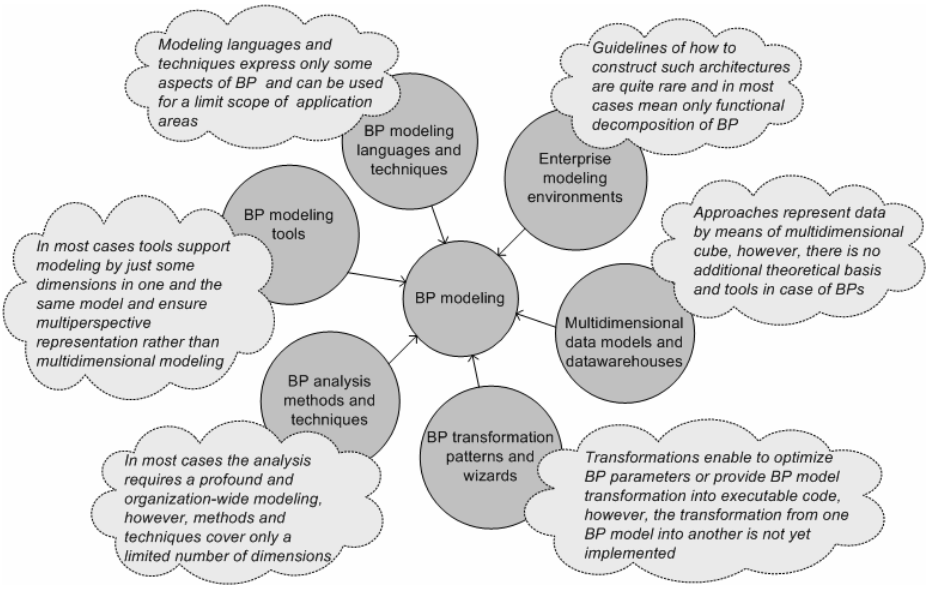


Fig. 1. Several disadvantages identified in areas relevant to BP modeling

Analyzing the situation, we can conclude that in several areas relevant to BP modeling there are some disadvantages (see Figure 1) that point to the necessity of multidimensional BP modeling approaches and environments. Analogous approaches for data analysis already exist, e.g., multidimensional data models and data warehouses have been developed to represent data by means of multidimensional cube [17]. However, there are no additional theoretical basis and tools in the case of BPs. Already available ingredients of BP modeling available in terms of methods, frameworks, tools and transformation algorithms should be integrated and expanded in order to develop comprehensive theoretical background for multidimensional BP modeling. This should provide an opportunity not only to view, but also to construct incrementally and

sensitively analyze BP dimension according to evolving business knowledge and multiple modeling purposes.

Multidimensional BP modeling should provide a normalized set of dimensions that are equally relevant in any BP modeling situation (see Figure 2 on the left), unlike essential BP modeling approaches that concentrate on certain aspects of BP and are applicable in a particular application area (see Figure 2 on the right). We regard dimension as a particular BP aspect or characteristic, where the complete set of dimensions allows illustrating any necessary characteristic of BP. Within multidimensional space it is possible to develop several process models as several variants of one and the same BP that are appropriate for particular modeling goal, e.g., developing BP maps for current and future business situation, analyzing BP improvement opportunities, developing new BPs, and developing requirements for new services. As a result, each of these models will reflect certain view or perspective of BP and include an appropriate set of dimensions. Here, we regard perspective as a way of showing or seeing from a particular position in order to achieve a pictorial representation or view of the entire MBPM model. The obtained view will represent part of the whole MBPM according to a particular modeling or analysis purpose.

The overall set of dimensions are interrelated in a particular way that makes it possible to switch between views in order to obtain personalized views on a MBPM (such as process role holder, process owner, process performer) and partially transform modeled in one model to another in order to extend MBPM with new information. It is essential that models at different dimensions may be obtained also from scratch or by reorganizing models initially developed from another viewpoint.

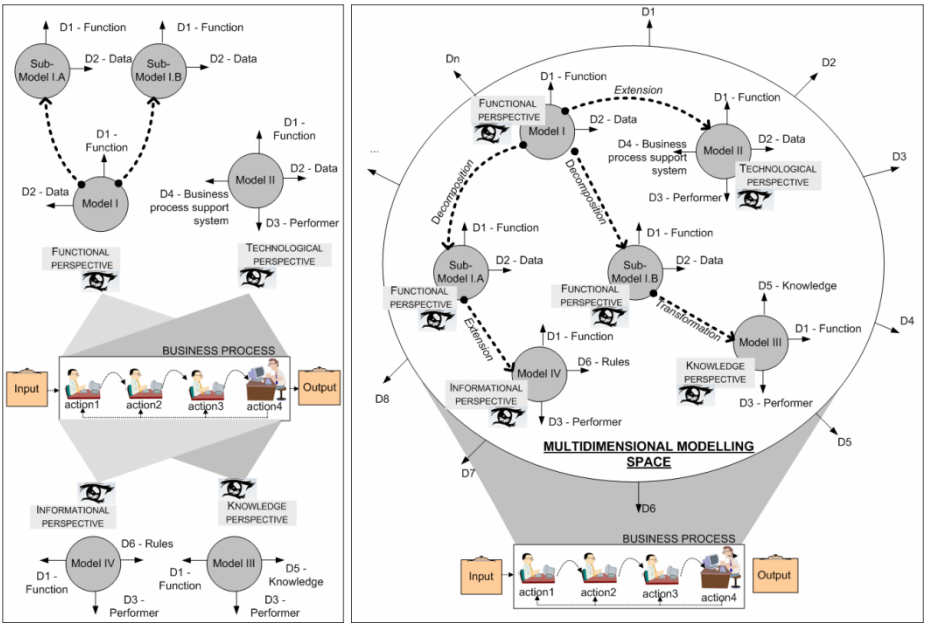


Fig. 2. Different BP modeling approaches: on the left - construction of the BP models in the classic way; on the right – BP modeling in the multidimensional modeling space

For example, we show a fragment of BP modeling using two different approaches: the classic approach in Figure 2 on the left and the multidimensional modeling approach in Figure 2 on the right. In both cases initial modeling purpose is to show functional decomposition of the process (Model I) reflecting two BP characteristics - functions and data. Further, the initial model is decomposed into two sub-models (sub-models I.A and I.B). Then, the modeling goal is changed and the technical aspects of the process are now in focus (Model II). Finally, models are extended with new dimensions that show informational perspective (Model III) and knowledge perspective (Model IV). In the case of classic modeling approaches the BP models are not related and are built either from scratch or one and the same model is extended with new dimensions. Alternatively in the case of multidimensional modeling approach, the obtained models are constructed incrementally extending or transforming initial models with necessary characteristics of BP that are relevant for current modeling situation. As a result, the obtained models are interrelated, e.g., business modeler or analyst can easily navigate through them by using different combinations of modeling dimensions in order to illustrate and analyze the desirable characteristics of the BP.

A description of the proposed approach and a research method that could be applied to develop the intended approach is given in sections below.

3 Research Approach

The research presented in the paper intends to develop an approach that combines a wide spectrum of BP characteristics in a common MBPM and allows the translation of BP model built from a particular perspective into other perspectives. The aim of the approach is to utilize the knowledge of the BP by its modeling in a multi-structured and systemic way and thus supporting BP analysis by multi-perspective BP representations. Expected results of the research are the constructs (concepts) and vocabulary of the domain, models and methods for the construction and analysis of MBPM that are experimentally tested in multiple domains of application, as well as requirements for the supporting modeling tool and its prototype.

The research process is divided into iterations according to design research framework [18]. The iteration is accomplished, if the planned results are achieved and the intermediate version of research is obtained. Each iteration includes following activities: 1) *Prepare* – define a questions for current iteration and plan investigation activities; 2) *Build* – make appropriate investigation, rationally interpret and implement the results ('implementation' can range from model to proof-of-concept prototype or to full implementation); 3) *Evaluate* – determine, if any progress have been made (basic question is how it works). Evaluation metrics and measurements should be developed; 4) *Theorize* – explicate the characteristics of the artifact and its operation with environment that result in the observed performance; 5) *Justify* – gather evidence to test the theory. In each iteration almost all of the mentioned tasks are carried out, the amount of work and granularity level only differs.

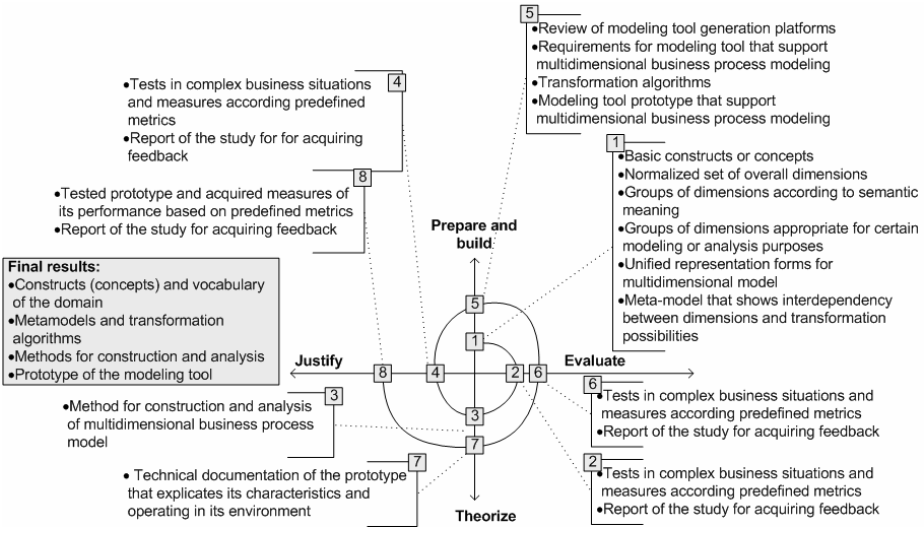


Fig. 3. The stages of scientific research

As the above-mentioned framework our research started with the investigation of relevant scientific literature in order to define basic constructs or concepts that form the vocabulary of domain, determine a normalized set of BP modeling dimensions, interdependency between dimensions, transformation possibility from one dimension into other, the available classifications and representation forms proposed by modeling techniques, languages and tools. As a result, we expect to build constraints, models and methods that should be evaluated (see Figure 3 1). Several reports of the study are to be prepared in order to receive feedback (see Figure 3 2). The next activity is theorizing that includes the development of the approach for multidimensional BP modeling and analysis (see Figure 3 3). The developed approach must be justified with appropriate metrics and the measurements, as well as tested in multiple domains of application, such as modeling of the processes of commercial and public organizations, knowledge-intensive processes, educational processes, and product development processes (see Figure 3 4). The next iteration (see Figure 3 5) starts with the investigation of modeling tool generation platforms that could be used for prototype development. Then, requirements and transformation algorithms could be proposed and the prototype developed and evaluated in a complex BP (Figure 3 6). The prototype should be described (Figure 3 7) and justified, int. al., the usability of given prototype should be compared to the usability of currently available advanced modeling tools (Figure 3 8).

The next section presents initial results of the first iteration that is not yet accomplished, e.g., the building activities of research (see Figure 3 1) are still under investigation. Further research activities, according given research framework, are discussed in Section 5.

4 Preliminary Results

In this section we discuss initial results of the doctoral work, that is: a) we specify general constructs such as "dimension", "perspective", "scale", "value" and "group of dimensions" for establishing theoretical basis of multidimensional modeling and analysis of BPs; b) point to the dimensions that are widely used for BP modeling tasks and are extensively discussed in scientific literature; c) define groups of dimensions based on semantic meaning; d) review presentation forms of modeling dimensions that provide some BP modeling tools; and e) outline basic stages (which are intended to be elaborated in future research) for constructing MBPM. A more detailed description of some results could be found in the following sub-sections.

A., B. and C. Basic constructs, widely used dimensions and groups of dimensions

We propose to base the notion of multidimensional BP model on the following multiple interrelated terms used in BP modeling discipline [15]:

- *Business process modeling dimension* – is a notion derived from mathematics where the dimension of space is roughly defined as the minimum number of coordinates needed to specify every point within it. In the case of BP modeling the dimension allows to represent a particular parameter associated with the BP, for example, time, process performer, information, etc. The finite set of all dimensions allows representing any set of relevant characteristics of BP, thus reflecting the real world in n-dimensional model (Figure 4).
- *Business process modeling perspective* – is a notion derived from graphical art where the perspective is a technique of representing the relationship or proportion of the parts of a whole, regarded from a particular standpoint or point of time. In the case of BP modeling a perspective is representation of the part of n-dimensional business process model, considered from a certain viewpoint (Figure 4). A perspective thus corresponds to a particular set of dimensions used in particular BP model representation.
- *Scale* – identifies the chosen granularity of a particular dimension, e.g., a time dimension scale can be a particular time period (Figure 4). The scale of a dimension may be reflected also as the hierarchy of decomposition/abstraction levels of a modeling aspect corresponding to the dimension (e.g., organizational hierarchy) [13].
- *Value* – identifies the denominations of each granule prescribed by the chosen scale of a dimension, e.g., the names of the days of the week or months for the time dimension (Figure 4).
- *Group of dimensions* – is a set of dimensions used simultaneously during the modeling process. An example of some groups of dimensions that could be found in related works [3, 14] is shown in Figure 4. Usually one of the dimensions is the "leading" dimension, which, e.g., is represented by swim lanes for grouping process model elements. The groups of modeling dimensions are candidates for the analysis perspectives.

It is essential to note that different granularity of a dimension may be required depending on the depth of analysis done from each perspective. The overall set of BP modeling dimensions that have their roots in popular enterprise architecture products

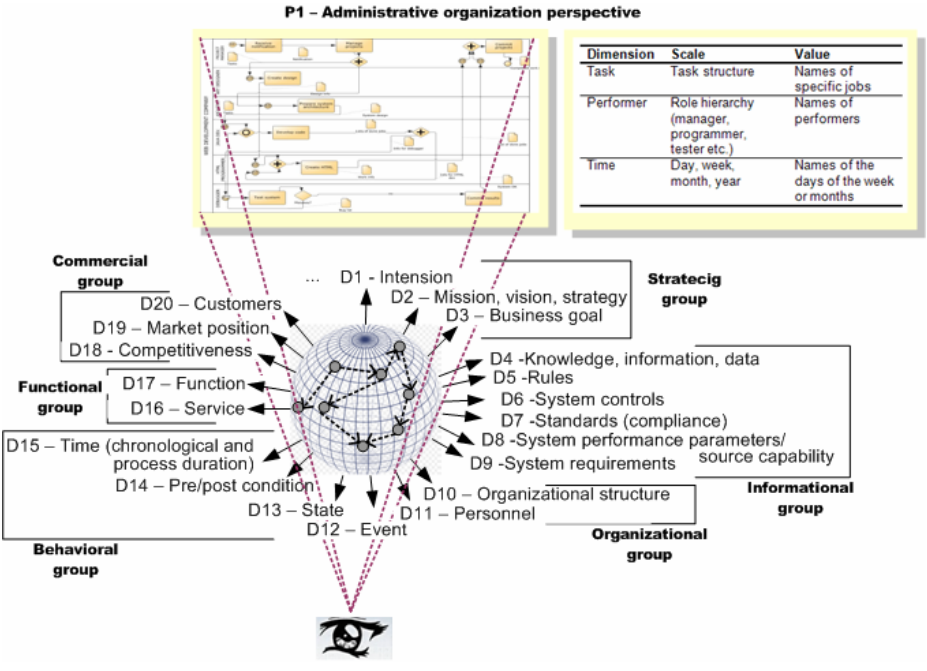


Fig. 4. Multidimensional space for BP modeling that consists of dimensions, groups of dimensions and perspectives (*for example, the administrative organization perspective includes three dimensions that have appropriate scale and values*)

and comparatively new non-traditional modeling frameworks and approaches, such as intentional, ontological, systems dynamics, ecosystems, and task-oriented oriented ones, as well as the possible grouping of dimensions based on the semantic meaning are discussed in our previous publications [13, 15].

D. and E. Basic construction stages and representation forms

The approach proposed in this paper takes a particular group of BP modeling dimensions as the starting point and then the BP model is gradually developed according to the needs of its analysis. Initially it contains a comparatively small amount of information and reflects one or a couple of dimensions. There are several approaches how to represent information according to particular dimensions of a chosen group of dimensions that are discussed in previous publications [13, 15]. Different approaches may be combined to achieve clarity and expressiveness of the model.

Subsequently the model may expand and be enriched with new dimensions, where some of the model parts could be automatically obtained from the already existing multidimensional model. We propose the following types of modifications [15]: *elaboration* – the existing model is modified without adding new dimensions, e.g., a new level of decomposition may be added; *extension* - the existing model is extended with new dimensions by reorganizing existing models initially developed according to other dimensions; *transformation* - the model is extended by new dimensions, some of which may be automatically obtained from the already existing part of the model. A more detailed description of construction stages could be found in [15].

5 Conclusions and Future Work

Preliminary results of the first year doctoral research are presented in this paper. An essential issue of constructing BP models, namely, the multidimensional space of process modeling was addressed. The purpose of the study was to analyze the motivation of and define the basic terms for multidimensional BP modeling, propose correspondence between different modeling dimensions and modeling dimension groups, define guidelines and provide an example of MBPM construction stages. The result of this paper reflects research in progress on BP modeling in multidimensional modeling space and points to the following future work:

- Provide guidelines suggesting what levels of scales and the range of values are to be used for each dimension and what canonic sets of dimensions are to be chosen for different modeling purposes
- Investigate each dimension separately in order to determine convenient ways of representing particular modeling dimensions and perspectives, and to develop unified form for representation of any dimension in the multidimensional modeling space
- Investigate the interdependence of dimensions and perspectives in order to define rules of transformations and rules of switching between the dimensions
- Define what purposes MBPM's are built for and what modeling techniques and languages are used for each modeling purpose
- Compare usability of multidimensional BP modeling to usability of currently available advanced BP modeling approaches by using Delphi method [16].

The use of the approach could facilitate related scientific researches, such as the analysis of BP flexibility and control possibility, the development of novel solutions for business and information systems in the context of enterprise collaboration networks, the development of service-oriented architecture for the support of BPs. Besides, it is expected that research results will be beneficial for commercial activities in the industry sector, as comprehensively designed BP may promote collaboration in enterprise networks in both organizational and technological contexts.

Acknowledgment

The author acknowledges scientific adviser of the doctoral thesis Dr.sc.ing. Marite Kirikova for valuable comments and suggestions on the draft of the paper and reviewers for their valuable comments on the initial version of the paper.

References

1. Fasbinder, M.: Why Model Business Processes? http://www.ibm.com/developerworks/websphere/library/techarticles/0705_fasbinder/0705_fasbinder.html
2. Brider, I.: Choosing Approach to Business Process Modeling - Practical Perspective (Inconcept issue 34) (January 2005), http://www.ibissoft.se/publications/Howto.pdf

3. de Bruin, B., Verschut, A., Wierstra, E.: Systematic analysis of business processes. Knowledge and Process Management 7(2), 87–96 (2000)
4. Havey, M.: Essential Business Process Modeling. O'Reilly, Sebastopol (2005)
5. List, B., Korherr, B.: An Evaluation of Conceptual Business Process Modelling Languages. In: SAC 2006, Dijon, France, April 23-27, pp. 1532–1539 (2006)
6. Ramachandran, B., Fujiwara, K., Kano, M., Koide, A., Benayon, J.: Business process trans-formation patterns & the business process transformation wizard. In: Proceedings of the 2006 Winter Simulation Conference, pp. 636–641 (2006)
7. GRADE, http://www.gradetools.com/default.htm
8. ARIS, http://www.ids-scheer.com/en/Software/ARIS_Software/3730.html
9. Pautasso, C., Alonso, G.: JOpera Visual Composition Language. Journal of Visual Languages and Computing (JVLC) 16(1-2), 119–152 (2005)
10. Li, L., Hosking, J., Grundy, J.: Visual Modelling of Complex Business Processes with Trees, Overlays and Distortion-based Displays. In: IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007), pp. 137–144 (2007)
11. WebSphere, http://www-01.ibm.com/software/websphere/
12. Jenz, D.E.: BPMO Tutorial: Defining a Private Business Process in a Knowledge Base, http://www.bpiresearch.com/BPMO_Tutorial.pdf
13. Kirikova, M., Businska, L., Penicina, L.: Multidimensional business process modelling. In: Proceedings of 7th International Conference on Perspectives in Business Informatics Research (BIR 2008), pp. 196–210 (2008)
14. Anohina, A., Grundspenkis, J.: Comparison of techniques for business process modeling. In: Scientific Proceedings of Riga Technical University, Computer Science. Applied Computer Systems. 5th series, vol. 8, pp. 80–91. RTU Publishing (2001)
15. Businska, L., Kirikova, M.: Multidimensional modeling and analysis of business processes. In: 13th East-European Conference on Advances in Databases and Information Systems (accepted for the print)
16. Linstone, H.A., Turoff, M. (eds.): The Delphi Method: Techniques and Applications (2002), http://www.is.njit.edu/pubs/delphibook/
17. Jensen, C.S., Kligys, A., Pedersen, T.B., Timko, I.: Multidimensional data modeling for location-based services. The VLDB Journal 13, 1–21 (2004)
18. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)

Author Index