Heinrich C. Mayr
Christian Kop
Stephen Liddle
Athula Ginige (Eds.)

LNBIP 137

Information Systems: Methods, Models, and Applications

4th International United Information Systems Conference
UNISCON 2012, Yalta, Ukraine, June 2012
Revised Selected Papers

Springer

Lecture Notes in Business Information Processing 137

Series Editors

Heinrich C. Mayr Christian Kop
Stephen Liddle Athula Ginige (Eds.)

Information Systems: Methods, Models, and Applications

4th International United Information Systems Conference
UNISCON 2012, Yalta, Ukraine, June 1-3, 2012
Revised Selected Papers

Springer

Volume Editors

Heinrich C. Mayr
Alpen-Adria-Universität Klagenfurt
Institute for Applied Informatics
Klagenfurt, Austria
E-mail: heinrich.mayr@aau.at

Christian Kop
Alpen-Adria-Universität Klagenfurt
Institute for Applied Informatics
Klagenfurt, Austria
E-mail: christian.kop@aau.at

Stephen Liddle
Brigham Young University
Provo, UT, USA
E-mail: liddle@byu.edu

Athula Ginige
University of Western Sydney
School of Computing, Engineering & Mathematics
Penrith, NSW, Australia
E-mail: a.ginige@uws.edu.au

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-38369-4 e-ISBN 978-3-642-38370-0
DOI 10.1007/978-3-642-38370-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013938054

ACM Computing Classification (1998): H.3, H.4, J.1, D.2

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

As with previous events, UNISCON 2012 again fulfilled its aim of being an international platform for researchers and IT professionals with an emphasis on popularizing the excellent research of Eastern European colleagues. To strengthen this aim, UNISCON 2012 not only was organized back-to-back with ICTERI 2012 but also exchanged highly rated papers according to the respective focuses of both conferences. UNISCON 2012 was held during June 1–3 at the Crimean State Humanitarian University in the beautiful surroundings of Yalta, Ukraine. There were 26 submissions of full papers to UNISCON 2012. Each paper was reviewed by at least three unbiased members of the Program Committee. In the end 10 full papers were selected for presentation and publication, yielding an acceptance rate of 38.5 %. ICTERI 2012 had 70 submissions, out of which 34 were selected for presentation at the conference and 14 for publication in the ICTERI proceedings, and four for the UNISCON proceedings. Thus, both conferences together had a full paper acceptance rate of 29.2 %. These proceedings represent the revised paper of a keynote talk, full papers, short papers, and extended poster abstracts that were finally submitted in November 2012. Providing these forms of submissions, the conference allowed for contributions of different stages of maturity, thus giving researchers a chance to present their work in progress and receive feedback from the audience.

The full papers were organized in six sessions (Data Management, Applications, Modeling and Semantics I, Modeling and Semantics II and Social Issues in Information Systems), each session containing two papers. In addition, four full papers were transferred from ICTERI (not published there), and organized in a separate session.

In his keynote address titled "Business Process Management: A Holistic Management Approach," Dimitris Karagiannis introduced existing methodologies on business process design.

In the Data Management Session, firstly Hendrik Decker argues in the paper "Maintaining Desirable Properties of Information by Inconsistency-tolerant Integrity Management" that data properties in information systems like quality, trustworthiness, certainty etc. can be modeled and maintained by database integrity technology. In the second paper of the session "Heuristic-Based Workload Analysis for Relational DBMSs," Andreas Lübcke, Veit Köppen, and Gunter Saake present heuristics based on workload information patterns to select the best relational database management system for a given workload.

Economic forecasting and decision-making information systems was the focus of the Applications Session. The paper "Information Technology for the Decision-Making Process in an Investment Company" by Tetiana Zakharova and Valentyna Moskalenko presents an information system for the analysis of investment markets. As the paper title "Long-Term Forecasting Technology of Macroeconomic Systems Development: Regional Aspect" already conveys, the article written by Marina Grinchenko, Olga Cherednichenko, and Igor Babych is devoted to the simulation of macroeconomic systems development. The proposed approach is based on the method of systems dynamics.

The Modeling and Semantics I session deals with quality in service-oriented software development and agile model-driven generation of web information systems. In the paper "Using Lightweight Activity Diagrams for Modeling and Generation of Web Information Systems," Dirk Reiß and Bernhard Rumpe describe how Web pages and workflow aspects are modeled using activity diagrams. They give an overview of the syntax and explain how the behavior is realized in the developed application. Vladimir A. Shekhovtsov, Heinrich C. Mayr, and Christian Kop ("Towards Conceptualizing Quality-Related Stakeholder Interactions in Software Development") address the interaction of business stakeholders and software developers in order to collect quality requirements.

The second session on Modeling and Semantics gives an inside look into ontology alignment and modeling in ambient assistant living. The paper "Agent-Based Implementation for the Discovery of Structural Difference in OWL-DL Ontologies" by Maxim Davidovsky, Vadim Ermolayev, and Vyacheslav Tolok proposes to use software agents for the effective use of knowledge-based systems in order to know the semantic relations between the ontologies that are used. The second paper, "Cognitive Modeling and Support for Ambient Assistance," by Judith Michael, Andreas Grießer, Tina Strobl, and Heinrich C. Mayr introduces the project HBMS and how specific modeling languages can be used to represent the individual behavior of an elderly person. These representations can be used for reproducing the cognitive model of a person if the person has forgotten some behavior.

Information systems can also empower social aspects. This is covered in the last UNISCON full paper session, "Social Issues in Information Systems." Athula Ginige, Tamara Ginige, and Deborah Richards contributed the paper "Architecture for Social Life Network to Empower People at the Middle of the Pyramid." They describe an architecture to enhance livelihood activities via the Internet. This architecture is intended for people who only possess mobile phones. Finally, the paper "Towards Quality Monitoring and Evaluation Methodology: Higher Education Case Study" by Olga Cherednichenko and Olga Yangolenko discusses problems of higher education monitoring.

The first paper of the ICTERI section discusses the validation of the OntoElect Methodology ("Validating OntoElect Methdology in Refining ICTERI Scope Ontology" by Olga Tatarintseva, Yuriy Borue, and Vadim Ermolayev). The second paper focuses on composition-nominative logics for software development ("Composition-Nominative Logics in Rigorous Development of Software Systems" by Mykola S. Nikitchenko and Valentyn G. Tymofieiev). The third paper presents an active data dictionary framework for information systems development ("Adaptable Enterprise Information Systems Development using Advanced Active Data Dictionary Framework" by Maxim Davidovsky, Gennadiy Dobrovolsky, Olga Todoriko, and Vladimir Davidovsky). The last paper proposes a model for the process of managing software variability ("The Model for Enhanced Variability Management Process in Software Product Line" by Olga Slabospitskaya and Andrii Kolesnyk).

The work-in-progress papers (short papers and extended abstracts) deal with the following topics:

– Web-Portals - "Providing the Interactive Studying of the Students which is Controlled by Distance with the Web-Portal E-olimp" by Borys Lyashenko, Sergiy Zhykovskyy, and Svitlana Postova.
– Object-Relational Transformations - "A Template-Based Method to Create Efficient and Customizable Object-Relational Transformation Components" by Igor Lihatsky, Anatoliy Doroshenko, and Kostiantyn Zhereb.
– Privacy in Virtual Worlds - "Are Anti-Goals Needed in Systems Design?" by Roland Kaschek
– Requirements Analysis - "Models and Tools for Multi-Dimensional Approach to Requirements Behavior Analysis" by Mykola Tkachuk and Irina Martinkus
– Data Warehouse Systems Architecture - "A Layered Architecture Approach for Large-Scale Data Warehouse Systems" by Thorsten Winsemann, VeitKöppen, Andreas Lübcke, and Gunter Saake
– Medical Information Technologies - "Towards Medical Screening Information Technology: The Healthgrid-Based Approach" by Karina Melnik, Olga Cherednichenko, and Vitaliy Glushko
– Requirements Engineering in AAL - "Knowledge-Oriented Approach to Requirements Engineering in Ambient-Assisted Living Domain" by Volodymyr Bolshutkin, Claudia Steinberger, and Mykola Tkachuk.

UNISCON 2012 was a success because of the effort and dedication of several people. First of all, the editors would like to thank all the authors mentioned above for their contributions. We would also like to thank the members of the Program Committee for their considerable reviews and the Springer team for their support in publishing these proceedings. Because of the quality of the contributions to these post-conference proceedings, which have been revised based on the reviews and discussions at the conference, we are sure the reader will find many interesting research results in here.

We also express our thanks and appreciation to our colleagues from the ICTERI Program Committee. It was a pleasure to co-operate in this joint adventure, which strengthened the quality of both conferences.

We would also like to thank the organization and relationship management team in Klagenfurt and Ukraine (in alphabetical order): Sergiy Bronin, Igor Grishin, Andreas Grießer, Yarova Ivanna, Denis Kasatkin, Stefan Leitner, Judith Michael, Heidi Scherzer, Klothilde Puschl, Tilmann Reuther, Christine Seger, Aljona Seko, and Tina Strobl. Without their support, UNISCON 2012 would not have been a success.

November 2012

Heinrich C. Mayr
Christian Kop
Stephen Liddle
Athula Ginige

Organization

UNISCON 2012 was organized by the department of Applied Informatics, Alpen-Adria-Universität.

Conference Chair

Heinrich C. Mayr	Alpen-Adria-Universität Klagenfurt, Austria

Program Co-chairs

Athula Ginige	University of Western Sydney, Australia
Stephen Liddle	Kevin and Debra Rollins Center for e-Business, USA

Program Committee

Witold Abramowicz	The Poznan University of Economics, Poland
Fahim Akhter	Zayed University, United Arab Emirates
Stephane Bressan	National University of Singapore
Ruth Breu	University of Innsbruck, Austria
Jörg Desel	Fernuniversität Hagen, Germany
Yogesh Deshpande	University of Western Sydney, Australia
Anatoly Doroshenko	Institute of Software Systems, Ukraine
Vadim Ermolayev	Zaporozhye National University, Ukraine
Günther Fliedl	Alpen-Adria-Universität Klagenfurt, Austria
Ulrich Frank	Universität Duisburg-Essen, Germany
Alexander Gelbukh	National Polytechnic Institute, Mexico
Marcela Genero	University of Castilla-La Mancha, Spain
Aditya Ghose	University of Wollongong, Australia
Athula Ginige	University of Western Sydney, Australia
Martin Glinz	University of Zurich, Switzerland
Mykola Glybovets	National University of Kyiv Mohyla Academy, Ukraine
Robert Goodwin	Flinders University, Adelaide, Australia
Igor Grishin	Crimean University for the Humanities, Ukraine
Jon Atle Gulla	Norwegian University of Science and Technology, Norway
Angappan Gunasekaran	University of Massachusetts, USA
Jean Luc Hainaut	University of Namur, Belgium

Wolfgang Hesse	Philipps University Marburg, Germany
Willem Jan-van den Heuvel	Tilburg University, The Netherlands
Jos van Hillegersberg	Erasmus University, The Netherlands
Jan Jürjens	Technische Universität Dortmund, UK
Gerti Kappel	Technical University of Vienna, Austria
Dimitris Karagiannis	Universität Wien, Austria
Bill Karakostas	City University, London, UK
Roland Kaschek	University of the Faroe Islands, Faroe Islands
Christian Kop	Alpen-Adria-Universität Klagenfurt, Austria
Myoung Ho Kim	KAIST, Korea
Kinshuk	Athabasca University, Canada
Mikhail R. Kogalovsky	Market Economy Institute RAS, Russia
Stephen Liddle	Kevin and Debra Rollins Center for e-Business, USA
Hui Ma	Victoria University of Wellington, New Zealand
Elisabeth Métais	Laboratory CEDRIC, Paris, France
Igor Nekrestyanov	St. Petersburg State University, Russia
Boris A. Novikov	St. Petersburg State University, Russia
Andreas Oberweis	Karlsruhe Institute of Technology (KIT), Germany
Maria E. Orlowska	The University of Queensland, Australia
Nataliya Pankratova	National Technical University Kyiv Polytechnic Institute, Ukraine
Òscar Pastor	Valencia University of Technology, Spain
Volodimyr Pasichnyk	National University Lvivska Polytechnika, Ukraine
Sudha Ram	University of Arizona, USA
Wolfgang Reisig	Humboldt-Universität zu Berlin, Germany
Matti Rossi	Helsinki School of Economics, Finland
Bernhard Rumpe	RWTH Aachen, Germany
Gunter Saake	Otto von Guericke University, Magdeburg, Germany
Klaus-Dieter Schewe	Software Competence Center Hagenberg, Germany
Joachim Schmidt	TU Hamburg Harburg, Germany
Vladimir Shekhovtsov	National Technical University KhPI, Kharkiv, Ukraine
Keng Siau	University of Nebraska-Lincoln, USA
Elmar Sinz	Universität Bamberg, Germany
Il-Yeol Song	Drexel University, Philadelphia, USA
Alexandr Spivakovsky	Kherson State University, Ukraine
Claudia Steinberger	Alpen-Adria-Universität Klagenfurt, Austria
Markus Stumptner	University of South Australia, Australia
Vijay Sugumaran	Oakland University, USA

Yuzura Tanaka	Hokkaido University, Japan
Bernhard Thalheim	Christian Albrechts University of Kiel, Germany
Helmut Thoma	IBM Switzerland, Switzerland
Mykola Tkachuk	National Technical University KhPI, Kharkiv, Ukraine
Farouk Toumani	Blaise Pascale University, France
Bhuvan Unkelhar	MethodsScience.Com, Australia
Tatjana Welzer	University of Maribor, Slovenia
Vyacheslav Wolfengagen	JurInfoR-MSU, Russia
Jian Yang	Macquarie University Sydney, Australia
Gregoriy Zholtkevych	V.N. Karazin Kharkiv National University, Ukraine
Joze Zupancic	University of Maribor, Slovenia

UNISCON Steering Committee

Athula Ginige	University of Western Sydney, Australia
Roland Kaschek	University of the Faroe Islands, Faroe Islands
Dimitris Karagiannis	University of Vienna, Austria
Stephen Liddle	Kevin and Debra Rollins Center for e-Business, USA
Heinrich C. Mayr	Alpen-Adria-Universität Klagenfurt, Austria
Mykola Tkachuk	National Technical University KhPI, Kharkiv, Ukraine

Organization Advisory Board

Alexander Bodnar	Director of the Scientific-Methodical Center for Distance Education of the Institute of Economics and Management, Crimean University of Humanities, Ukraine
Sergiy Bronin	Ukraine Association of IT Professionals, Kiev, Ukraine
Mykola Ignatenko	First Vice Rector of the Crimean University for the Humanities, Ukraine
Rena Larina	Director of the Institute of Economics and Management of the Crimean University for the Humanities, Ukraine
Olena Ponomaryova	Vice-Rector on scientific work of the Crimean University for the Humanities
Yuriy Smelyanski	Vice-Head on educational work, Institute of Economics and Management of the Crimean University for Humanities, Ukraine

Organizing Committee

Head: Sergiy Bronin	Ukraine Association of IT Professionals, Kiev, Ukraine
Igor Grishin	Crimean University for the Humanities, Ukraine
Andreas Grießer	Alpen-Adria Universität Klagenfurt, Austria
Yarova Ivanna	Ukrainian association of IT professionals
Denis Kasatkin	Company "Region," Simferopol, Ukraine
Stefan Leitner	Alpen-Adria-Universität Klagenfurt, Austria
Judith Michael	Alpen-Adria-Universität Klagenfurt, Austria
Heidi Scherzer	Alpen-Adria-Universität Klagenfurt, Austria
Klothilde Puschl	Alpen-Adria-Universität Klagenfurt, Austria
Christine Seger	Alpen-Adria-Universität Klagenfurt, Austria
Aljona Seko	Crimean University for the Humanities, Ukraine

Austrian-Ukrainian Relationship Management

Tilmann Reuther	Alpen-Adria Universität Klagenfurt, Austria
Mykola Tkachuk	National Technical University KhPI, Kharkiv, Ukraine

Honory Committee

Vladimir Semynozhenko	Chairman of the National Agency on Science, Innovation and Informatization of Ukraine
Vitalina Dzoz	Minister of Education, Science, Youth and Sports of the Autonomous Republic of The Crimea
Olexander Gluzman	Rector of Crimean University for the Humanities, Ukraine
Valeriy Kosarev	Head of the Education, Science, Youth and Sports Constant Commission, the Supreme Rada(Council) of the Autonomous Republic of Crimea
Leonid Tovazhniansky	Rector of the National Technical University Kharkov Polytechnical Institute, Ukraine

Sponsoring Institutions

BOC Group, Austria

Table of Contents

Social Issues in Information Systems

Selected ICTERI Papers

Short Papers

Extended Abstracts

Business Process Management: A Holistic Management Approach

Dimitris Karagiannis

Research Group Knowledge Engineering, University of Vienna
Bruenner Straße 72, 1210, Vienna, Austria
dk@dke.univie.ac.at

Abstract. BPM is a holistic management approach applied by organizations worldwide in different settings and scenarios to address complex requirements of their stakeholders. In this paper an introduction on existing methodologies on business process design and tools is provided and the BPMS paradigm as a generic life cycle approach is proposed for application of a meta modelling platforms to address current and future challenges in different BPM related settings. Categorization of the best practices according to the application context is provided by presenting corresponding cases.

Keywords: Business Process Management, Meta Models, Modelling Methods, BPMS, Compliance, Competence, e-Government, Continuous Improvement.

1 Introduction on BPM

The Business Process Management (BPM) is a well established and widely – cross-domain – applied holistic management approach, which can be defined as a set of *structured methods and technologies* [1] for managing and transforming *organizational operations* [2]. If we consider the organizational life cycle, where each company goes from initial start-up, over (rapid) growth toward maturity and decline, it can be observed that in the growth and maturity phase (according to [3]) the need for structuring and formalizing policies, rules and procedures rises. Therefore, companies acting globally – hence having exponential growth of the complexity of involved stakeholders, market and law regulations, etc – have to be flexible and to have their procedures and workflows well planned.

This requires management concepts capable of providing methods to (1) design working procedures and optimize/diminish internal and external interfaces and (2) enable an IT system supporting the required flexibility – the BPM. As we can see there is a scenario-imposed division between the drivers for the application of the BPM. On one side coming from the strategy corner, business trends such as globalization, dynamic markets, fast information transfer, global standards, inter-organizational value chains, quick decision making and outsourcing, and on the other hand coming from the IT corner we have flexible IT architectures, mobile IT and new means of communication. This driver-based segregation results in two so-called BPM camps: the Business driven BPM – e.g. BPM lifecycle, Business Re-engineering and

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 1–12, 2013.

IT driven BPM – e.g. Standard Software - ERP, Workflow Management Systems (WfMS) and Service Oriented Architectures (SOA).

Historically the BPM as an approach has been applied in and characterized by different settings and scenarios – e.g. it's' ancestors described by A. Smith in [4]. In the modern era the first period (1900-1950) was characterized by Industrial Engineering by Ford [5], Taylor's Scientific Management [6] and Statistical Quality Control by Shewhart [7]. Next period (1950-1980) was influenced by the works in the domain of System Theory [8] and System Dynamics [9] and findings in the manufacturing domain – e.g. Toyota [10] and General Motors [11]. In the 80's the Total Quality Management [12], Value Chain [13], Six Sigma [14] and the Zachmann Framework [15], continued shaping the BPM. In the 90's we distinguish two periods, first, the early years were influenced by the works in the domain of Continuous Improvement [16], IBM's LOVEM Methodology [17], Business Process (Re)engineering [18], Architecture of Integrated Information Systems (ARIS) [19], Process Innovation [20] and findings resulting from Workflow Management Coalition undertakings [21]. Second period, the late 90's pushed the BPM toward approach presented in this paper. This period was characterized by Business Process Management Systems - BPMS [22], Integrated Definition Methods [23], Unified Modelling Language - UML and Business Process Management Initiative [24].

The aforementioned phases in the history can also be viewed based on the so called BPM traditions. Namely if we start from early 20th century and move toward present time, we can identify following phases (as outlined in [25]): the work simplification phase – ending with WWII production, the first computers, PC; globalization phase afterward and third phase characterized by outsourcing and rise of the internet. The two phases starting after the work simplification phase – in early 80's toward present time, can be mapped to the previously mentioned advances in BPM over time and viewed based on their tradition aspect:

(1) The Quality Control Tradition – focusing on introducing and expanding aspects in conjunction with Total Quality Management (building on a premise that all members of an organization focus their efforts to meet the customers quality expectations, and thus enhance the processes and products), Six Sigma (focus is on identifying and removing the causes of errors and standardizing manufacturing and business processes) and Lean Manufacturing (identifying and removing all processes and activities that have different goals than producing value for the end user – so-called waste reduction as outlined in [10]) resulting in so-called Lean Six Sigma as well as in Maturity Models (e.g. CMMI, BPMM as detailed in section 2.1)

(2) The Management Tradition – starting from Porters Value Chain (definition of the value adding activities in the process chain) and Rummler-Brache Performance Improvement (identifying the nine performance variables on the organizational, process and performer level addressing the goals, design and management on each level – see [16] for details) in 80's toward BP Re-Engineering, Balance Scorecards, Process Frameworks covering different scenarios such as SCOR for managing supply chain concerned with Plan (demand and supply management), Source (acquiring resources based on plan), Make (finalization of the product/service), Deliver (transportation and delivery management) and Return (post delivery support) processes, eTOM for managing business processes in ICT and entertainment domains

– focusing on three major areas – a.) Strategy, Infrastructure and Product, b.) Operations and c.) Enterprise Management; and considering Business Process Architectures.

(3) The Information Technology Tradition – the more IT focused part of the BPM considering IT (prominent example is the Zachman Framework currently available in its third version – [15]) and Enterprise Architectures (e.g. FEAF by US federal government – providing reference models for: Performance (PRM), Business (BRM), Service Component (SRM), Data (DRM) and Technical (TRM) which are applied by the federal government to align their operations), Case Tools, Business Process Modelling Tools and Methods, Business Rules and Business Intelligence, Expert Systems, Workflows and also the BPMS.

(4) The Organizational Tradition – as the name says is focused on the organizational aspects of the BPM – spanning from actual organization design theories as outlined in [56], toward management of cultural changes and aspects [34] and organizational transformations. [57] provides one example of a modelling language/framework – FCEO – which considers several organizational entities required to enable design of the organizational structures.

In addition to the aforementioned views on the BPM, the important aspect as argued by [25] and based on the involvement of the organization implementing the BPM is the co called organizational viewpoint. The survey conducted from 2005 until 2009 [28] brought into a view fact that although prominently placed the absolute number of organization who see BPM as “a top down methodology designed to organize, manage and measure the organization based on the organization’s core processes” was reduced by 9% as opposed to organizations who see BPM as “a systematic approach to analyzing, redesigning, improving and managing a specific process” that rose by approximately 20%. It is also interesting that Europeans still favour the first definition whereas North Americans favour the latter and that in the preliminary survey results for 2011, stakeholders changed their opinion again and perceived again the BPM as “ …a top-down...” – see [55] for details. When we picture the BPM, the generic view almost always includes: (1) Business Processes, (2) Organization, (3) Services and Products and (4) Information Technology. In this paper we would like to focus on (1) and (2). Using this generic view we define the *business process* as a *sequence of activities*, which are executed by *actors* to create *products* by working on *artefacts* and by using *resources*. Authors in [26] envision following types of business processes based on their flexibility/structure: (1) Ad-Hoc (short lived, urgent processes – e.g. completely new task), (2) Weakly Structured (integrated and open processes, cooperation activities – e.g. reporting in a project), (3) Structured (structured processes such as life insurance application). The business processes, if we go back to our definition, are executed by systems or humans structured in different organization types. It can be observed that, in companies utilizing the BPM, in the long run there is a change or a shift from hierarchical organizational structure toward more process oriented structure having defined process responsibilities. According to [27] the process oriented structure involves more than 100 roles connected to the application of BPM. Categorization introduced in [28] distinguishes: (1) Business roles – such as business analyst, etc, (2) IT roles – e.g. system architect and (3) Operational roles – e.g. process owner, process performer. Based on the actual activity performed by one of the roles they may be involved in different phases of the

BP life cycles. After this introductory section, setting the view on the BPM, in the following section, an overview on BP life cycles is provided and enriched by proposal on meta BP life cycle – the BPMS [22]. This section finalizes with an overview on BP design and BP technologies that can support the BPMS meta life cycle. In the third section five selected cases in the area of e.g. competence management, etc. are presented. Paper concludes with a summary and presents some of the challenges that BPM will face in the near future.

2 BPM Frameworks

As outlined previously the BPM has been applied in an organized and structured way in various organizations and settings, yielding results impacted by constraints imposed by different methodologies (cmp. [12] and [14]). Combination of a specific procedure to introduce the BPM, selecting an appropriate tools and technologies and executing different design patterns resulted in best practices extending the initial BPM framework into different categories as outlined in [28], [29], [34], [47] and utilized by different suites [59]. Section 2.1 provides an overview on some of the available life-cycles and presents one of them – following a holistic approach to BPM application – the BPMS. Section 2.2 discusses the design of the BP with a venture toward meta-modelling approaches. And finally section 2.3 provides a brief outlook on technologies applied nowadays to make use of the BPM.

2.1 The BPMS Life Cycle

The general concept of a Life Cycle is understood slightly different based on the application scenario. For example [29] defines it as "…*dynamic, iterative process of changing the enterprise over time by incorporating new business processes*…" for the EA domain, on the other hand [30] defines the life cycle as "…*the series of changes that the members of a species undergo as they pass from the beginning of a given developmental stage to the inception of that same developmental stage in a subsequent generation*…" in the biology domain. Accordingly in the BPM domain different views and implementations of the BP Life Cycle exist, cmp. [31], [32], [33], [34], etc. based on the methodology, application scenarios and preferences of the users. Without implying to have a faultless solution, from a generic point of view it can be observed that most of them are to an extent covered by following phases: (1) Process Strategy, (2) Process Design, (3) Process Implementation, (4) Process Execution and (5) Process Controlling. In order to support organizations going through these phases we mapped this generic view to the BPMS paradigm [22] to provide answers to these questions: (1) Which *products* do we offer?, (2) How do we design our *business processes*?, (3) How do we *operate* our business processes?, (4) How do we *control* our daily business?, (5) How do we *evaluate* our business? The phases or processes of the BPMS (as outlined in [35]), that can be directly mapped to the aforementioned questions, include:

(1) Strategic Decision Process – targeting definition of strategic and general conditions, success factors and essential criteria for business processes. This phase is

carried out after a strategic decision has been made concerning the requirement for reengineering of an enterprise's organisational environment. It takes into account many different aspects such as: culture of the enterprise, general and specific project prerequisites for success indicators, product and process portfolio, etc.

(2) Re-Engineering Process – concerned with documentation, adaptation, modelling and optimization of business processes, identification of reorganization possibilities and capacities. The primary objective is the modelling and redesign of the business process taking into account evaluation criteria from Strategic Decision Process. Techniques applied in redesign to reach the desired quality of business process include e.g. simulation, analysis, etc.

(3) Resource Allocation Process – focused on implementation of business processes based on IT or organizational issues, assignment of technical and human resources for execution. Resource Allocation Process takes responsibility over correct identification and coordination of resources required for execution. It may involve modification of such resources – e.g. virtualisation of legacy applications.

(4) Workflow Process – dealing with execution of business processes in operational environment, gathering operational data for further analysis and evaluation. Reengineered business process is executed using resources made available in the Resource Allocation Process. Execution generates the necessary information for the Performance Evaluation Process – to perform the evaluation.

(5) Performance Evaluation Process – tackling aggregation and processing of business process and organizational data, extraction of measurements and metrics. The primary objective is the qualitative and quantitative evaluation of the information acquired from execution of the re-designed BP, and continuous feedback to Strategic Decision and Re-Engineering Process.

In the BPMS the aforementioned processes are dynamically and iteratively executed in a bootstrapping approach both in each phase (e.g. running more than once) as well as in the whole BPMS. The BPMS paradigm, as outlined here, is a method independent framework providing support from strategic management to the operational business process execution allowing continuous control and adaptation of the company goals without focusing on a single BP or business level but on a global view. Additional important aspect interrelated with BP Life Cycles are the Maturity Models in BPM. Most prominent examples include the Capability Maturity Model Integration (CMMI) [36] and Business Process Maturity Model (BPMM) from OMG [37]. Based on the degree of involvement in the organization and time [38] distinguishes following phases in the evolution of the Maturity Models: (1) - Application Development e.g. - CMMI, SW-CMM, (2) IT Organization e.g. - IT-MM, ITIL, COBIT, (3) Business Sector e.g. - P-MM, ISO, MBNQA, and (4) Process Management Maturity Model e.g. PM-MM, BPMM. An approach to combine the BP Lifecycle and Maturity Models is outlined in [39] providing a mapping to the Process Management Life Cycle (PMLC).

2.2 The Design Task

Within the phases of the BPMS different tasks are run in order to satisfy the input/output requirements of the selected phase. Some tasks are prominently applied across the BPMS for yielding different results, e.g. in all phases we use Design Task to design different issues (e.g. Strategy models in Strategic Decision Process, Resource models in Resource Allocation Process, etc.). The task of business process design is a process where modelling is performed and evaluated (e.g. through analysis, simulation, prototyping) with a goal to generate a business process as an end result. Intention to perform such activity as business process design, is given in requirement to be able to describe the process, to analyze the process and to be able to enact it. [32] defines the process modelling as "*…the representation of the business process in terms of a business model using a process language…*". Modelling as a concept refers to the process of mapping the reality to a model based on the analysis and structuring of the available data material and models are seen as "*representation of either reality or vision*" [40], representing the real world in an agreed syntax and semantics. When speaking of models we can basically distinguished between non-linguistic or iconic models that use signs and symbols that have an apparent similarity to the concepts of the real world and linguistic models that use basic primitives such as signs, characters or numbers. Nearly all models in computer science are of the latter linguistic type. Linguistic models can be further distinguished in being realized with textual and graphical / diagrammatic languages [41]. The four layer meta model architecture (original - model - meta model - meta² model) as detailed in [42] depicts that on the next layer (after the designed model) we have the modelling language (ML). The ML is defined by syntax, semantics and notation that provide the necessary modelling primitives in order to build the model. The concepts that describe the ML are defined in the meta meta model language. ML can be classified according to paradigm they follow [43]: (1) Graph-based languages – processes are seen as sequences of activities, (2) Rule-based languages – processes are seen as complex systems, (3) Speech act-based languages – processes are seen as interactions of at least two stakeholders and (4) System-dynamic based languages – processes are seen as systems taking account also dependencies between e.g. employee motivation, trust, product quality, etc. The other classification (according to [44]) follows the application scenarios, e.g.: (1) Traditional process ML – such as IDEF, EPC, Petri Nets, (2) Object-oriented ML - presenting the problem domain (modelling the business) but focusing on the solution domain (modelling the software) - e.g. UML AD, (3) Dynamic process ML - such as WPDL, BPMN and (4) Process integration ML - having focus on integrating the processes of two or more business partners - e.g. ebXML. Additional aspects in this context are also the different views on the business process model, e.g.: (1) Functional – e.g. Activity, Sub process, (2) Dynamic – e.g. Control Flow, Information Flow, (3) Organizational– e.g. Actor, Resource, (4) Content – e.g. Product, Artefact, (5) Quantitative – e.g. Times & Costs, Probabilities & Stat. Distributions, and (6) Time-Oriented – e.g. Version, Variant. We use different methods to evaluate the designed business processes- e.g. by simulation, empirical studies, etc. The evaluation represents one of the core activities in the design of business processes – analysis, simulation and prototyping play an important role. Usually combination of these three evaluation mechanisms is carried out to evaluate a business process design: In the first step the business process model is analyzed according to its structure, behaviour, etc. and based on the findings the model is redesigned accordingly (e.g. in different model versions). These design alternatives are in the subsequent step simulated and based on the results of the simulation the business process model is adapted. Finally based on the outcomes of the simulation the design is approved or in an additional step an organizational prototyping is done to further evaluate and select the most promising design alternative. The Business Process Analysis (BPA) advantages lie in the fact that the analysed model does not have to be modelled strictly formal and provides quite fast an overview on the optimization alternatives, but it should be combined with other two mentioned mechanisms to deliver better results. When using BPA to perform structural correctness assessment, according to [45], following techniques can be used: (1) Verification – to assess if system is free of logic errors (e.g. livelocks) and context independent and (2) Validation – to assess if system behaves as expected. The Business Process Simulation (BPS), performs a complementary check to BPA as it concentrates on the dynamic aspects of the business process design, namely it performs a dynamic evaluation of process covering times (e.g. cycle time, execution time, waiting time…) and the workload. BPS allows assessment of the re-engineering activities before actual implementation / execution, but on the other hand requires higher effort in modelling and acquisition of timings beforehand. Business Process Prototyping is concerned with deployment and testing of the business process design in the real-life settings. In contrast to BPS, this mechanisms allows testing of other factors than time and workload – e.g. quality of output, but on the other hand, the efforts connected with setting up the environment and actual execution of an organizational prototype are much higher. Additional aspects to be considered when performing the business process design are the layers of business process models and the roles/addresses concerned with them. Layers include: (1) Process Landscape layer – concerned with management processes (managing the core and support processes), core processes (interaction with external environment – e.g. customers) and support processes (supporting the employees internally in execution of the core processes). Processes on this layer are addressed by CEO, etc. (2) Business Process Layer – concerned with As-Is BP of the enterprise, roles involved on this layer include: Process Responsible, Business Process Expert, etc. (3) Workflow layer - targeting technical To-Be processes – addressing roles of IT Specialist, System Architect, etc. (4) Microflow layer – technical details, e.g. UML sequence Diagram – addressing Application Developers, etc. Model transformation is applied in business process design when it is required to integrated models situated on different layers and/or between different modelling languages. This is done either by Model2Model – so called horizontal transformation where both source as well as target is a model; or in the so-called vertical transformation where we have Model2Text, Text2Model (e.g. Model to Documentation or Code) or Roundtrip scenario – in case of synchronization.

Business Process design is also characterized by application of different standards as depicted in [46], for example for graphical standards – BPMN, UML AD, natural language standards – SBVR, interchange standards – XPDL, BPDM, execution standards – BPEL, diagnosis standards – BPRI, BPQL, etc. In the following section a brief overview on the technologies applied to realize BPM is outlined.

2.3 The Technology View

Design and execution of business processes is carried out utilizing a specific set of business process tools. Based on the survey conducted by [59] stakeholders in the business process management domain apply BPM tools for: (1) Optimization of processes, (2) Increased productivity for process workers, (3) The ability to model business process (4) Support for compliance efforts, (5) Standardize processes across divisions and regions, (6) The ability to provide real-time visibility into key processes, (7) The ability to change processes quickly and easily. Based on the application scenario specifics there are many different categorization approaches that can be applied to the BPM tools (cmp. [28]), but in general they can be divided in tools that primarily support the management of the business/enterprise architecture – so called Business Process Analysis Tools and tools that also offer functionality to execute the business processes – so called Business Process Management Tool suites.

Business Process Analysis (BPA) Tools are used to manage the Business/Enterprise Architecture based on business processes with focus on organization, roles, documents, resources, IT and their interdependencies in a holistic way - so the main target is the design, analysis, simulation and optimization of the business processes and not on their execution. Tools in this category include: BOC Group (ADONIS), Software AG (ARIS), Casewise (Corporate Modeller), etc. as outlined in [28]. The Business Process Management Tools (BPMT) provide the ability to design and execute business processes, mostly based on the service-oriented architecture paradigm with the goal of co-called ‘zero-code programming‘. Their main focus is on the actual execution of the business process. Some major providers for BPM Suites (as in [59]) are: IBM (IBM WebSphere), Software AG (webMethodsBPMS), Oracle (Oracle BPM Suite) etc.

If we try to map the tools toward our generic BP life cycle and the BPMS accordingly, we can say that BPA tools support Process Design, to an extend the Process Strategy and share the Process Implementation with BPMT, where they focus on Process Execution and partly to Process Controlling. Another important aspect when considering Business Process tools is their environment used to realize and utilize the tools. On the architecture side, tools cover range from standalone application to client-server installation, lately also involving BPM in the Cloud. On the realization side, some tools profit from the advanced functionalities provided by the platforms they were built upon, e.g. samples presented in the section 3, utilize the advanced meta modelling functionalities provided by the ADOxx® [48] to extend the applied Modelling Method (modelling language, modelling procedure and mechanisms & algorithms). As outlined in [48], ADOxx® platform is a meta modelling-based development and configuration environment to create domain-specific modelling tools, that is, it provides the ability to extend the meta model of the modelling languages utilized in a BP tool to cover scenario specific or end user based requirements. According to [49] and [59] the trends in the Business Process Software Market go toward (1) the work on business process getting more social/collaborative dimension, (2) the higher impact of the BPM SaaS offerings, lowering the barriers for new clients to get started with the business process management, (3) Business processes and data quality are joined at the hip and (4) BPMN 2.0 improvements of the communication across the process teams. Additionally the revival of the Dynamic Case Management focusing on more dynamic, unstructured, ad hoc, untamed processes is predicted by [49]. Next section provides samples of scenarios realized following the described life cycle (section 2.1) and utilizing BP tools described in this section.

3 BPM: Selected Cases

BPM constitutes the basis for different application scenarios and strategic activities. The BPMS [22] is a method independent framework providing support from strategic management to the operational business process execution allowing continuous control and adaptation of the company goals without focusing on a single BP or business level but on a global view. ADOxx® is a meta modelling platform for development and configuration of domain-specific modelling tools and can be used to extend existing modelling methods and standards. There exist different categorization possibilities for the applications of the BPMS, and in this section our focus is on the following perspectives: (1) human – e.g. competence management, (2) domain – e-Government, (3) continuous improvement – quality management, (4) functional – activity based costing, (5) and regulatory – compliance management.

In the Competence Management (CM) the main driver is to support a Process driven CM (as outlined in [52]). Namely the CM is an integrated management concept directly interrelated with Intellectual Capital Management (CM is used to manage the Intellectual Capital), Human Resources Management (CM is providing requirements to HR), Business Process Management (CM receives requirements and provides improved processes) and Knowledge Management (CM is defining knowledge needs), therefore the goal was to develop a PCM modelling method capable of combining a person-centred and a process-centred view on competences and supporting a holistic competence lifecycle. PCM extended the general BPM method with CM related concepts and modelling classes such as Competence Catalogue, Global Job Profiles but also extending existing classes such as Company Map with CM related attributes. More details on the conceptualization and implementation of the PCM modelling method can be found in [53].

Interesting solutions, utilizing the vehicles presented here, have also been applied in the e-Government domain, in order to support highly dynamic environment (different stakeholders, changes affecting many processes – e.g. law amendment, etc) a hybrid approach has been applied to combine a BPM based modelling method with a Business Rules approach, and to externalize the dynamic part of the e-government processes to rules and thus reduce the complexity when adapting deployed processes. Details on this approach, conceptualization of the modelling method, application and execution of the processes can be found in [54].

In the Functional Perspective, an interesting approach is the application of the Activity Based Costing based on [57] to the BPM where the where the focus is set on defining the actual process cost as starting point for the internal cost allocation, and external cost calculation through definition of a transparent activity based costing which is defined in the extended attributes of the modelling objects.

In the Continuous Improvement the focus is in most cases on integrating different methodologies [10], [12], [14] in the applied BPM framework in order to cover measurement and evaluation and continuous improvement of the business process performance. This, based on the select approach, may focus on the improvement of process concerned with direct-customer value generation, or by focusing on improving applicable organizational processes.

In the Compliance Management setting the main challenge is the requirement of extending the business processes with the regulatory context. Initial step is the consideration of the regulatory context to derive the requirements – e.g. sources to be considered included (1) actual legal regulations – e.g. SOX – the Sarbanes-Oxley Act of 2002, (2) best practice available – e.g. COSO Framework, ITIL as a reference, and (3) expert knowledge – e.g. auditors and risk experts knowledge. This analysis results in defining the functional requirements (Design Environment – focusing on the modelling capabilities; Execution/Testing Environment – focusing on the interaction capabilities; Analysis Environment – focusing on the controlling capabilities) and non-functional requirements (Role Distinction/Involvement and Organisational procedure for SOX compliance - the stepwise approach). The realization followed extension of the standard BPM modelling method with concepts for risk management. Namely using the meta modelling functionalities offered by ADOxx®, the modelling classes (e.g. Activity, Document, etc.) were extended with SOX modelling classes Account, Category, Control, Control Objective and Event. Second the BPM relation classes were extended with SOX specific relation classes – e.g. includes (for setting relations between ERM elements), and third references between the BPM and ERM elements were defined. Please see [50] and [51] for more detailed description on the compliance management presented here.

4 Summary and Future Challenges

This paper provided an overview on BPM, its challenges and current developments. Main focus was on the life cycles, proposing a generic life cycle – the BPMS, for managing business processes without focusing on a single business process scenario or business level but on a global view and also introduced an approach on how, based on the end user requirements and scenario specifics, meta modelling platforms (here ADOxx®) can be used to address the requirements by extending existing and conceptualizing new modelling methods in selected cases from the BPM domain. It is obvious that in the future the challenges for the described approaches will be in the direction of adding more weight to the collaborative and distributed BPM, paying attention also on the mobile aspects of today's world and by enabling hybrid approaches on amalgamating the existing BPM methods in highly complex scenarios. Many of these challenges are currently being addressed by the scientific and industrial communities at the Open Models Initiative [58], which provides an open approach to sophisticated platform enabling conceptualization, development, deployment and application of a domain specific modelling methods.

References

1. Bandara, W., et al.: Business Process Management Education in Academia: Status, Challenges, and Recommendations. Communications of the Association for Information Systems 27, Article 41 (2010)
2. Hammer, M.: What is Business Process Management? In: Handbook on Business Process Management, vol. 1 (2010)
3. Daft, R.L.: Organizational Theory and Design. West Publishing, St. Paul (1992)
4. Smith, A.: An Inquiry into the Nature and Causes of the Wealth of Nations. Management Laboratory Press (2008) (First publication: 1776)
5. Ford, H.: Discuss Americas Industrial Future (1894)

6. Taylor, F.W.: The Principles of Scientific Management. Harper & Brothers Publishers, New York & London (1911)
7. Shewhart, W.A.: Statistical method from the viewpoint of quality control. Courier Dover Publications (1986) (First publication: 1931)
8. Bertalanffy, L.V.: General System Theory: Foundations, Development, Applications, Revised Edition. George Braziller, Inc. (1969)
9. Forrester, J.W.: Principles of Systems, New edition. Productivity Press (1988) (First publication: 1968)
10. Ōno, T.: Toyota production system: beyond large-scale production. Productivity Press (1988) (First publication: 1978)
11. Sloan, A.P.: My years with General Motors. Doubleday (1964)
12. Deming, W.E.: Out of the crisis. MIT Press (2000) (First publication: 1982)
13. Porter, M.E.: Competitive Advantage: Creating and Sustaining Superior Performance, 1. Aufl. Free Press (1998) (First publication: 1985)
14. Tennant, G.: SIX SIGMA: SPC and TQM in Manufacturing and Services. Gower Publishing, Ltd. (2001)
15. Zachman, J.A.: A framework for information systems architecture. IBM Systems Journal (1987)
16. Rummler, G.A., Brache, A.P.: Improving Performance: How to Manage the White Space on the Organization Chart. Jossey-Bass Inc., U.S (1990)
17. Soper, P.J.A., et al.: Business Process Reengineering and Beyond. IBM, Redbooks (1995)
18. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution, Rev Upd. Harper Paperbacks (2004) (First publication: 1993)
19. Scheer, A.: Architecture of Integrated Information Systems: Foundations of Enterprise Modelling, 1st edn. Springer (1994) (First publication: 1991)
20. Davenport, T.H.: Process innovation: reengineering work through information technology. Harvard Business Press (1993)
21. Workflow Management Coalition – WfMC, `http://www.wfmc.org/`
22. Karagiannis, D.: BPMS: Business Process Management Systems. ACM SIGOIS, Special Issue: Business Process Reengineering 16(1) (1995)
23. Mayer, R.J., et al.: IDEF3 Process Description Capture Method Report, KBSI-IICE-90-STR-01-0592-02 (1995)
24. Business Process Management Initiative (BPMI) – 1999, now Business Modelling and Integration DTF, `http://www.bpmi.org`
25. Harmon, P.: The Scope and Evolution of Business Process Management. In: Handbook on Business Process Management, vol. 1. Springer (2010)
26. Nastansky, L., et al.: The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automation. In: 13th WCC. Springer (1994)
27. Association of Business Process Management Professionals: Leitfaden zum Prozessmanagement Body of Knowledge ABPMP BPM CBOK® (2009)
28. Wolf, C., Harmon, P.: A BP Trends Study -The State of Business Process Management (2010)
29. Chief Information Officer Council: A Practical Guide to Federal Enterprise Architecture (2001)
30. Life Cycle. Encyclopædia Britannica Online (2012) (retrieved April 25, 2012)
31. Scheer, A., Brabänder, E.: The Process of Business Process Management. In: Handbook on Business Process Management, pp. 239–265. Springer (2010)
32. Weske, M., van der Aalst, W.M.P., Verbeek, H.M.W.: Advances in business process management. Data & Knowledge Engineering 50 (2004)
33. Houy, C., Fettke, P., Loos, P.: Empirical research in business process management – analysis of an emerging field of research. BPMJ 16(4) (2010)

34. vom Brocke, J.: The Six Core Elements of Business Process Management. In: Handbook on Business Process Management, vol. 1. Springer (2010)
35. Karagiannis, et al.: Introduction to Business Process Management Systems Concepts, Business Process Modelling, Scholz-Reiter, B., Stickel, E. (Hrsg.). Springer (1996)
36. CMMI, `http://www.sei.cmu.edu/cmmi`, (retrieved April 25, 2012)
37. OMG, BPMM, `http://www.omg.org/spec/BPMM` (retrieved April 25, 2012)
38. Ho, D.: Business process maturity model – A CMM-based business process reengineering research, Stevens Institute of Technology (2012)
39. PMLC, `http://www.pmlc.at/docs/BOC_PMLC_folder_web_de.pdf`
40. Peters, R.J., Ozsu, M.T.: Reflection in a Uniform Behavioral Object Model. In: Elmasri, R.A., Kouramajian, V., Thalheim, B. (eds.) ER 1993. LNCS, vol. 823, pp. 34–45. Springer, Heidelberg (1994)
41. Harel, D., Rumpe, B.: Modeling Languages: Syntax, Semantics and All That Stuff (2000)
42. Karagiannis, D., Kühn, H.: Metamodelling platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, p. 182. Springer, Heidelberg (2002)
43. Junginger, S.: Modellierung von Geschäftsprozessen - State-of-the-Art, neuere Entwicklungen und Forschungspotenziale, PhD Thesis, University of Vienna (2000)
44. Mili, H., et al..: Business process modeling languages: Sorting through the alphabet soup. ACM Computing Surveys (CSUR) 43, 4:1–4:56 (2010)
45. van der Aalst, W.M.P.: Making Work Flow: On the Application of Petri Nets to Business Process Management. In: Esparza, J., Lakos, C.A. (eds.) ICATPN 2002. LNCS, vol. 2360, pp. 1–22. Springer, Heidelberg (2002)
46. Ko, R.K.L., Lee, S.S.G., Lee, E.W.: Business process management standards: a survey. BPM Journal 15(5), 744–791 (2009)
47. Karagiannis, D., Woitsch, R.: Knowledge Engineering in Business Process Management. In: Handbook on Business Process Management, vol. 2. Springer (2010)
48. Kühn, H.: The ADOxx® Metamodelling Platform. In: Workshop Methods as Plug-Ins for Meta-Modelling in Conjunction with Modellierung 2010, Klagenfurt, Austria (2010)
49. Le Clair, G., Miers, D.: The Forrester WaveTM: Dynamic Case Management, Q1 2011. Forrester Research, Inc. (2011)
50. Karagiannis, D., Mylopoulos, J., Schwab, M.: Business Process-Based Regulation Compliance: The Case of the Sarbanes-Oxley Act. In: 15th IEEE RE (2007)
51. Karagiannis, D.: A Business Process-Based Modelling Extension for Regulatory Compliance. Multikonferenz Wirtschaftsinformatik (2008)
52. Berio, G., Harzallah, M.: Knowledge Management for Competence Management. In: Proceedings of I-KNOW, Austria, pp. 154–162 (2005)
53. Hintringer, S., Nemetz, M.: Process driven Competence Management: A Case Study at Hilti Corporation. In: WI 2011, Austria (2011)
54. Hrgovcic, V., et al.: Adaptive and Smart e-Government Workflows - Experience Report from the Projects FIT and LDCAST, eChallenges, Sweden (2008)
55. Wolf, C., Harmon, P.: A BP Trends Study -The State of Business Process Management (2012)
56. Berente, N., Claggett, J.: Challenges in the Management of High-Performance Computing Centers: An Organizational Perspective. In: SC 2011, USA (2011)
57. Staubus, G.J.: Activity Costing and Input-Output Accounting (1971)
58. Open Models Initiative (2012), `http://www.openmodels.at`
59. Forrester Research, Forrester WaveTM: BPM Suites Q3 (2010)

Maintaining Desirable Properties of Information by Inconsistency-Tolerant Integrity Management

Hendrik Decker*

Instituto Tecnoloógico de Informática, Universidad Politécnica de Valencia, Spain
hendrik@iti.upv.es
http://web.iti.upv.es/~hendrik/

Abstract. We argue that several kinds of desirable semantic properties of data provided by information systems, such as their quality, trustworthiness, certainty, security or lack of riskiness, can be modeled and maintained by existing database integrity technology. As opposed to the usual requirement for conventional integrity constraints, formulas that model such properties may not always be perfectly satisfied. Therefore, it is necessary to be less rigorous about the satisfaction of such properties. That can be achieved by recently introduced techniques of inconsistency tolerance, in the fields of integrity checking, integrity repair and query answering. We support our argument by an extended example.

Keywords: quality, uncertainty, security, integrity, inconsistency, tolerance.

1 Introduction

In information systems, first-order logic sentences called *integrity constraints*, or simply *constraints*, express conditions required to be invariantly satisfied across state changes caused by updates of the underlying database. Logic also serves to assert other semantic information that may transgress the tabular structures of databases. For instance, desirable properties about the security, trustworthiness or reliability of data, or the undesirability of negative features such as uncertainty, riskiness or hazardousness, can be modeled by constraints.

Once modeled as constraints, such properties can be maintained by methods that normally are used for checking or repairing integrity. For checking the quality of data, that has been shown in [13]. Data that lack desirable properties may also impair the quality of answers to queries. As we are going to see, the approach in [10] to answers that have integrity (abbr. *AHI*) in inconsistent databases can be generalized to reliable query answering in information systems with data that possibly are unreliable in any kind of way.

* Supported by FEDER and the Spanish grants TIN2009-14460-C03, TIN2010-17139.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 13–24, 2013.

Maintaining desirable properties modeled by constraints is based on the idea that the stored data comply with the properties if the database satisfies the respective constraints. Conversely, the amount of constraint violations can be considered as a measure of the shortcomings of the current contents of the database.

However, the semantics of sentences that model arbitrary properties may be different from that of integrity constraints. Thus, the use of integrity checking for maintaining such properties may seem questionable. In fact, conventional integrity checking requires that all constraints are totally satisfied before updates are checked for integrity preservation. However, requiring total satisfaction of arbitrary, though desirable properties tends to be overly strict, since they may well be infringed without hampering ongoing operations. Fortunately, many integrity checking methods are *inconsistency-tolerant*, i.e., they do not have to insist on the mentioned requirement of total satisfaction, since they are able to evaluate constraints correctly in the presence of manifest logical inconsistency [14].

In this paper, we are going to show how to gain a systematic control over possible imperfections of the data provided by information systems, by expressing desirable properties as constraints, maintaining them by inconsistency-tolerant methods for checking and repairing integrity, and querying them by providing answers that are not unacceptably deficient with respect to the constraints.

In Section 2, we illustrate how desirable properties can be captured in the syntax of integrity constraints. In Section 3, inconsistency-tolerant integrity checking (abbr. *ITIC*) is shown to be appropriate for preventing the deterioration of desirable properties of stored data. In Section 4, *ITIC* is shown to be usable also for repairing damaged properties of data, without insisting on a total elimination of all violations, as conventional inconsistency repair does. In Section 5, we sketch how the quality of answers to queries containing impaired data can be assessed and accepted even if their causes are involved in constraint violations. In Section 6, we feature an extended example of managing data that may have the undesirable property of being uncertain, with comparisons to alternative approaches. In Section 7, we address related work. In Section 8, we conclude.

The framework of this paper essentially consists of well-known *datalog* formalisms and terminology. [1]. Moreover, let symbols like D, I, IC, U always stand for a database, an integrity constraint, a finite set of constraints (also called *constraint theory*) and, resp., an update. The result of updating D by U be denoted by D^U, and $D(S)$ be the result of evaluating a set of sentences S in D.

2 Modeling Desirable Properties as Constraints

We are going to exemplify that desirable properties of data such as their security, trustworthiness and reliability, or the absence of uncertainty, riskiness or hazardousness, can be modeled in the syntax of integrity constraints.

Constraints usually are asserted as first-order sentences, and often as *denials*, i.e., clauses with empty head of the form $\leftarrow B$, where the body B is a conjunction of literals that expresses what *should not* be true in any state of the database. If a constraint I expresses what *should* be true, then a denial form of I can be obtained by re-writing $\leftarrow \sim I$ in clausal form, as outlined, e.g., in [9]. Instead of leaving the head of denial constraints empty, it is also good practice to put there a predicate that expresses some lack of soundness. For instance, $violated \leftarrow B$ explicitly states that each instance of B that is true in the database is not in accordance with the integrity required by the formula $\leftarrow B$.

An example of a constraint that should hold: in a medical database,

$$\forall x \forall y \forall z(cholesterol(x, y, z) \rightarrow person(x) \wedge mmol\text{-}per\text{-}l(y) \wedge time(z))$$

stipulates that the first attribute of the *cholesterol* table always is a person, the second a millimoles/liter value and the third of type *time*. Similarly, the denial

$$\leftarrow cholesterol(x, y1, z),\ cholesterol(x, y2, z),\ y1 \neq y2$$

declares a primary key constraint on the first and third columns, preventing multiple entries for a person with different cholesterol values at the same time.

Likewise, conditions for characterising database entries as risky, suspicious or precarious information can be modeled in the syntax of integrity constraints. For instance, consider the denial $\leftarrow risk(x, z)$, where *risk* is defined by

$$risk(x, z) \leftarrow person(x),\ cholesterol(x, y, z),\ above\text{-}threshold(y)$$

and $above\text{-}threshold(y)$ be a predicate that compares the value of y with a suitable constant boundary. Thus, the health of a person x at time z is considered to be at risk if the cholesterol level y exceeds some critical value.

An example of a constraint that states doubts about the validity of data in a municipal information system for, say, administering voters in some election is

$$dubious \leftarrow birth\text{-}date(x, z),\ z < \mathit{1900}$$

by which each entry of a person x with birth date z before the 20th century is characterised as dubious, i.e., not trustable. Also sentences such as

$$uncertain(x) \leftarrow confidence(row(x, y), z) \wedge z < th$$

which amalgamate meta-predicates about data structures into first-order terms [4], may serve as constraints for disqualifying the trustworthiness of rows x in database tables y whose confidence value z is below some threshold th (which may either be a constant parameter or the result of some evaluable function).

Another example for showing the syntactic uniformity of constraints that either model integrity or other desirable properties is given by the two sentences

$$violated \leftarrow email(x),\ sent(x,y),\ received(x,z),\ y > z \quad (\star)$$

which signals an integrity violation if the sent-date y of an email is after its received-date z (assuming normalized time zones for y and z), and

$$suspect(x) \leftarrow email(x, from(y)),\ \sim authenticated(y) \quad (\star\star)$$

which warns that an email x sent by y is suspect if y cannot be authenticated. The violation of $(\star)$ indicates an inconsistency that should not occur, while, for violations of $(\star\star)$, x may well be acceptable despite suspicions caused by y.

Conventionally, updates that would violate any regular integrity constraint either are rejected or its violations should be repaired, in the process of updating or some time soon thereafter. As opposed to that, the preceding examples illustrate that violations of constraints describing desirable properties, such as $\leftarrow suspect(x)$, may be tolerable, even for extended periods of time, some more, some less. In the remainder, we are going to show that all kinds of constraints can be maintained by inconsistency-tolerant approaches to the management of integrity, for checking the preservation of integrity by updates, or repairing violations, or providing answers that have integrity in databases that don't.

3 Inconsistency-Tolerant Constraint Checking

The integrity constraints of a database are meant to be checked upon each update, which is committed only if it does not violate any constraint. The same can be done for any other desirable property that has the form of a constraint.

However, total integrity is hard to be maintained throughout the entire lifetime of a database, and the violation of other desirable properties may at times be acceptable anyway, as we have seen in Section 2. Thus, methods for checking constraints that can tolerate extant violations are needed.

In [14], we have formalized ITIC, and have shown that many (though not all) existing integrity checking methods are inconsistency-tolerant, although most of them have been designed to be applied only if integrity is totally satisfied before any update is checked.

As seen in [11], most integrity checking methods can be described by *violation measures*, which are a form of inconsistency measures [17]. Each such measure maps pairs (D, IC) to some partially ordered space, for sizing the violated constraints in (D, IC). Thus, an update can be accepted if it does not increase the measured amount of constraint violations.

Definition 1 defines each constraint checking method $\mathcal{M}$ (in short, method) as an I/O function that maps input triples (D, IC, U) to $\{ok, ko\}$. The output ok means that the checked update is acceptable, and ko that it may not be acceptable. For deciding to ok or ko an update U, $\mathcal{M}$ uses a violation measure μ, the range of which is structured by some partial order $\preccurlyeq$, for determining if U increases the amount of measured violations or not.

Definition 1. (*Measure-based ITIC*)
An integrity checking method maps triples (D, IC, U) to $\{ok, ko\}$. For a violation measure $(\mu, \preccurlyeq)$, a method $\mathcal{M}$ is *sound* (resp., *complete*) for μ-based integrity checking if, for each (D, IC, U), (1) (resp., (2)) holds.

$$\mathcal{M}(D, IC, U) = ok \quad \Rightarrow \quad \mu(D^U, IC) \preccurlyeq \mu(D, IC) \tag{1}$$

$$\mu(D^U, IC) \preccurlyeq \mu(D, IC) \quad \Rightarrow \quad \mathcal{M}(D, IC, U) = ok \tag{2}$$

The only real difference between conventional methods and integrity checking as defined above is that the former additionally requires total integrity before the update, i.e., that $D(IC) = true$ in the premise of Definition 1. The measure μ used by conventional methods is binary: $\mu(D, IC) = true$ means that IC is satisfied in D, and $\mu(D, IC) = false$ that it is violated.

As seen in [14], many conventional methods can be turned into sound (though not necessarily complete) inconsistency-tolerant ones, simply by waiving the premise $D(IC) = true$ and comparing violations in (D, IC) and (D^U, IC). If there are more or new violations in (D^U, IC) that are not in (D, IC), then they output *ko*; otherwise, they may output *ok*. Hence, we have the following result.

Theorem 1. For each (D, IC, U) where IC is a set of desirable properties modeled as a constraint theory, and each violation measure μ, the violations of IC in D and D^U can be compared by each sound μ-based integrity checking method.

4 Inconsistency-Tolerant Repairs

Essentially, repairs are updates of databases that eliminate their constraint violations. However, the user or the application may not be aware of each violation, so that some of then may be missed when trying to repair a database.

Below, we recapitulate the definition of repairs in [14] which is inconsistency-tolerant since it permits that some violations may persist after a partial repair.

Definition 2. (*Repair*)
For a triple (D, IC, U), let S be a subset of IC such that $D(S) = false$. An update U is called a *repair* of S in D if $D^U(S) = true$. If $D^U(IC) = false$, U is also called a *partial repair* of IC in D. Otherwise, if $D^U(IC) = true$, U is called a *total repair* of IC in D.

Unfortunately, partial repairs may inadvertently cause the violation of some constraint that is not in the repaired subset.

Example 1. Let $D = \{p(1,2,3),\, p(2,2,3),\, p(3,2,3),\, q(1,3), q(3,2), q(3,3)\}$ and $IC = \{\leftarrow p(x,y,z) \wedge {\sim}q(x,z),\ \leftarrow q(x,x)\}$. Clearly, both constraints are violated. $U = \{\mathit{delete}\, q(3,3)\}$ is a repair of $\{\leftarrow q(3,3)\}$ in D and hence a partial repair of IC. It tolerates the persistence of the violation $\leftarrow p(2,2,3) \wedge {\sim}q(2,3)$ in D^U. However, U also causes the violation $\leftarrow p(3,2,3) \wedge {\sim}q(3,3)$ of the first constraint of IC in D^U. That instance is not violated in D. Thus, the non-minimal partial repair $U' = \{\mathit{delete}\, q(3,3),\ \mathit{delete}\, p(3,2,3)\}$ is needed to eliminate the violation of $\leftarrow q(3,3)$ in D without causing a violation that did not yet exist.

Although U' does not cause any unpleasant side effect as U does, such repair iterations may in general continue indefinitely, as known from repairing by triggers [20] [6]. However, that can be alleviated by checking if a given repair is an update that *preserves integrity*, i.e., does not increase the amount of violations, with any measure-based method that prevents inconsistency from increasing while tolerating extant constraint violations. Hence, we have the following result.

Theorem 2. Let μ be a violation measure, $\mathcal{M}$ a μ-based integrity checking method and U a partial repair of IC in D. For a tuple (D, IC), U preserves integrity wrt. μ if $\mathcal{M}(D, IC, U) = ok$.

For computing partial repairs, any off-the-shelve view update method can be used, as follows. Let $S = \{\leftarrow B_1, \ldots, \leftarrow B_n\}$ be a subset of constraints to be repaired in the database D of an information system. Candidate updates for satisfying the view update request can be obtained by running the view update request *delete violated* in $D \cup \{violated \leftarrow B_i \mid 0 \le i \le n\}$. For deciding if a candidate update U is a valid repair, U can be checked for integrity preservation by some measure-based method, according to Theorem 2. More details about the computation of partial and total repairs can be found in [12].

5 Answers That Tolerate Violations of Desirable Properties

Violated constraints that model desirable properties may impair the integrity of query answering, since the data that provide the answers are possibly insecure or precarious. Thus, an approach to provide answers with integrity in databases with violated constraints is needed.

Consistent query answering (abbr. *CQA*) [2] provides answers that are true in each minimal total repair of IC in D. CQA uses semantic query optimization [7] which in turn uses integrity constraints for query answering. A similar approach is to abduce consistent hypothetical answers, together with a set of hypothetical updates that can be interpreted as integrity-preserving repairs [15].

A new approach to provide answers that have integrity (abbr. AHI) is presented in [10]. AHI determines two sets of data: those by which an answer is deduced, i.e., the causes of the answer, and those that cause constraint violations. Each cause is a subset of *iff*(D) [11], i.e., all ground instances of if- and only-if halves of predicate completions in *comp*(D) [8]. An answer θ has integrity if the intersection of one of the causes of the answer with the causes of constraint violations is empty, since θ is deducible from data that are independent of those that violate constraints. A more precise definition in [10] is summarized below.

Definition 3. Let θ be an answer to a query in D. We say that θ *has integrity in* (D, IC) if there is a cause C of θ in D such that $C \cap C_{IC} = \emptyset$, where C_{IC} is the union of all causes of constraint violations in (D, IC).

AHI is closely related to measure-based ITIC, since some convenient violation measures are defined by causes: cause-based methods accept an update U only if U does not increase the number or the set of causes of constraint violations [11]. Similar to ITIC, AHI is inconsistency-tolerant since it provides correct results in the presence of constraint violations. However, AHI is not as inconsistency-tolerant as measure-based ITIC, since each answer accepted by AHI is independent of inconsistent parts of the database, while measure-based ITIC may admit updates that violate constraints. For instance, U in Example 1 causes the violation of a constraint while eliminating some other violation. Now, if U is checked by a method based on a measure that assigns a greater weight to the eliminated violation than to the newly caused one, U will be *ok*-ed, because it decreases the measured amount of inconsistency.

In fact, it is possible to provide answers that, despite some tolerable degree of contamination with inconsistency, are appreciable. We sketch a relaxation of AHI in the definition below. Its objective is to provide answers that tolerate a certain amount of inconsistency that may be involved in the derivation of answers. To quantify that amount, some application-specific tolerance measure τ (say) is needed. In ongoing research, we elaborate a theory based on Definition 4.

Definition 4. Let τ be a tolerance measure that maps elements in the powerset of *iff*(D) to a space that is structured by some partial order $\preceq$. Let *th* be a threshold value in that space up to which violations are tolerable. Then, an answer θ to some query in D is said to *tolerate violations up to th* if there is a cause C of θ such that $\tau(C \cap C_{IC}) \preceq th$, where C_{IC} is as in Definition 3.

6 Uncertainty Management – An Example

Uncertainty of data often cannot be totally avoided. Hence the desire to contain or reduce the uncertainty of data, so that they do not worsen over time nor compromise the integrity of answers too much. In this section, we illustrate how to meet that desideratum by inconsistency-tolerant integrity management, and discuss some more conventional alternatives. In particular, we compare inconsistency-tolerant integrity management with brute-force constraint evaluation, conventional integrity checking that is not inconsistency-tolerant, total repairing, and CQA, in 6.1–6.6.

The example below is open to interpretation. By assigning convenient meanings to predicates, it can be interpreted as a model of uncertainty in a decision support systems for, e.g., stock trading, or controlling operational hazards in a complex machine.

Let D be a database with the following definitions of view predicates *ul*, *um*, *uh* that model uncertainty of low, medium and, respectively, high degree:

$$ul(x) \leftarrow p(x, x)$$
$$um(y) \leftarrow q(x, y),\ \sim p(y, x)\ ; \quad um(y) \leftarrow p(x, y),\ q(y, z),\ \sim p(y, z),\ \sim q(z, x)$$
$$uh(z) \leftarrow p(0, y),\ q(y, z),\ z > th$$

where *th* be a threshold value that always is greater or equal 0. Now, let uncertainty be denied by the following constraint theory:

$$IC = \{\leftarrow ul(x), \ \leftarrow um(x), \ \leftarrow uh(x)\}.$$

Note that IC is satisfiable, e.g., by $D = \{p(1, 2), p(2, 1), q(2, 1)\}$. Now, let the extensions of p and q in D be populated as follows.

$$p(0, 0),\ p(0, 1),\ p(0, 2),\ p(0, 3), \ldots, p(0, 1000000),$$
$$p(1, 2), p(2, 4), p(3, 6), p(4, 8), \ldots, p(500000, 1000000)$$
$$q(0, 0),\ q(1, 0),\ q(3, 0),\ q(5, 0),\ q(7, 0), \ldots, q(999999, 0)$$

It is easy to verify that the low-uncertainty denial $\leftarrow p(x, x)$ is the only constraint that is violated in D, and that this violation is caused by $p(0, 0)$.

Now, let us consider the update U = *insert* $q(0, 999999)$.

6.1 Brute-Force Uncertainty Management

For later comparison, let us first analyse the general cost of brute-force evaluation of IC in D^U. Evaluating $\leftarrow ul(x)$ involves a full scan of p. Evaluating $\leftarrow um(x)$ involves access to the whole extension of q, a join of p with q, and possibly many lookups in p and q for testing the negative literals. Evaluating $\leftarrow uh(x)$ involves a join of p with q plus the evaluation of possibly many ground instances of $z > th$.

For large extensions of p and q, brute-force evaluation of IC clearly may last too long, in particular for safety-critical uncertainty monitoring in real time. In 6.2, we are going to see that it is far less costly to use an ITIC method that simplifies the evaluation of constraints by confining its focus on the data that are relevant for the update.

6.2 Inconsistency-Tolerant Uncertainty Management

First of all, note that the use of conventional simplification methods that require the satisfaction of IC in D is not allowed in our example, since $D(IC) = \mathit{false}$. Thus, conventional integrity checking has to resort on brute-force constraint evaluation. We are going to see that inconsistency-tolerant integrity checking of U is much less expensive than brute-force evaluation.

At update time, the following simplifications of medium and high uncertainty constraints are obtained from U. (No low uncertainty can be caused by U since $q(0, 999999)$ does not match $p(x, x)$.) These simplifications are obtained at hardly any cost, by simple pattern matching of U with pre-simplified constraints that can be compiled at constraint specification time.

$$\leftarrow \sim p(999999, 0)\ ; \quad \leftarrow p(x, 0),\ \sim p(0, 999999),\ \sim q(999999, x)$$
$$\leftarrow p(0, 0),\ 999999 > th$$

By a simple lookup of $p(999999, 0)$ for evaluating the first of the three denials, it is inferred that $\leftarrow um$ is violated.

Now that a medium uncertainty has been spotted, there is no need to check the other two simplifications. Yet, let us do that, for later comparison in 6.3.

Left-to-right evaluation of the second simplification essentially equals the cost of computing the answer $x = 0$ to the query $\leftarrow p(x, 0)$ and successfully looking up $q(999999, 0)$. Hence, the second denial is *true*, which means that there is no further medium uncertainty. Clearly, the third simplification is violated if $999999 > th$ is true, since $p(0, 0)$ is true, i.e., there possibly is a high uncertainty.

Now, let us summarize this subsection. Inconsistency-tolerant integrity checking of U according to Theorem 1 essentially costs a simple access to the p relation. Only one more look- up is needed for evaluating all constraints. And, apart from a significant cost reduction, ITIC prevents medium and high uncertainty constraint violations that would be caused by U if it were not rejected.

6.3 Inconsistency-Intolerant Uncertainty Management

ITIC is logically correct, but, in general, methods that are not inconsistency-tolerant (e.g., those in [16,18]) are incorrect, as shown by the example below.

Clearly, p is not affected by U. Thus, $D(\leftarrow ul(x)) = D^U(\leftarrow ul(x))$. Recall that each method that is not inconsistency-tolerant assumes $D(IC) = true$. Thus, such methods would wrongly conclude that the unfolding $\leftarrow p(x,x)$ of $\leftarrow ul(x)$ is satisfied in D and D^U, although $p(0,0) \in D$. That conclusion is then applied to $\leftarrow p(0,0),\ 999999 > th$, (the third of the simplifications in 6.2), which thus is taken to be satisfied in D^U. That, however, is wrong if $999999 > th$ is true. Thus, non-inconsistency-tolerant integrity checking may wrongly infer that the high uncertainty constraint $\leftarrow$ *uh(z)* cannot be violated in D^U.

6.4 Uncertainty Management by Repairing (D, IC)

Conventional integrity checking requires $D(IC) = true$. To comply with that, all violations in (D, IC) must be repaired before each update. However, such repairs can be exceedingly costly, as argued below.

In fact, already the identification of all violations in (D, IC) at update time may be prohibitively costly. But there is only a single low uncertainty constraint violation in our example: $p(0,0)$ is the only cause of the violation $\leftarrow$ *ul(0)* in D. Thus, to begin with repairing D means to request $U = \textit{delete}\ p(0,0)$, and to execute U if it preserves all constraints, according to Theorem 2.

To check U for integrity preservation means to evaluate the simplifications

$$\leftarrow q(0,0) \quad \text{and} \quad \leftarrow p(x,0),\ q(0,0),\ {\sim}q(0,x)$$

i.e., the two resolvents of ${\sim}p(0,0)$ and the clauses defining *um*, since U affects no other constraints. The second one is satisfied in D^U, since there is no fact matching $p(x,0)$ in D^U. However, the first one is violated, since $D^U(q(0,0)) = true$. Hence, also $q(0,0)$ must be deleted. That deletion affects the constraint

$$um(y) \leftarrow p(x,y),\ q(y,z),\ {\sim}p(y,z),\ {\sim}q(z,x)$$

and yields the simplification

$$\leftarrow p(0,y),\ q(y,0),\ {\sim}p(y,0).$$

As is easily seen, this simplification is violated by each pair of facts of the form $p(0,o)$, $q(o,0)$ in D, where o is an odd number in $[1, 999999]$. Thus, deleting $q(0,0)$ for repairing the violation caused by deleting $p(0,0)$ causes the violation of each instance of the form $\leftarrow um(o)$, for each odd number o in $[1, 999999]$.

Hence, repairing each of these instances would mean to request the deletion of many rows of p or q. We shall not further track those deletions, since it should be clear already that repairing D is complex and tends to be significantly more costly than ITIC. Another advantage of ITIC: since inconsistency can be temporarily tolerated, ITIC-based repairs do not have to be done at update time. Rather, they can be done off-line, at any convenient point of time.

6.5 Uncertainty Management by Repairing (D^U, IC)

Similar to repairing (D, IC), repairing (D^U, IC) also is more expensive than to tolerate extant constraint violations until they can be repaired at some more convenient time. That can be illustrated by the three violations in D^U, as identified in 6.1 and 6.2: the low uncertainty that already exists in D, and the medium and high uncertainties caused by U and detected by ITIC. To repair them obviously is even more intricate than to only repair the first of them as tracked in 6.4.

Moreover, for uncertainty management in safety-critical applications, it is no good idea to simply accept an update without checking for potential violations of constraints, and to attempt repairs only after the update is committed, since repairing takes time, during which an updated but unchecked state may contain possibly very dangerous uncertainty of any order.

6.6 Reliable Constraint Evaluation in Uncertain Databases

As already mentioned in Section 5, CQA is an approach to cope with constraint violations for query evaluation. There is a clear kinship of ITIC and CQA, since checking and repairing uncertainty constraints involves their evaluation. However, the evaluation of constraints by CQA is unprofitable, since consistent answers are defined to be those that are true in each minimally repaired database. Thus, by definition, CQA returns the empty answer for each queried denial constraint I, indicating the satisfaction of I. Thus, answers to queried constraints computed by CQA have no meaningful interpretation.

For example, CQA computes the empty answer to the query $\leftarrow ul(x)$ and to $\leftarrow uh(z)$, for any extension of p and q. However, the only reasonable answers to $\leftarrow ul(x)$ and $\leftarrow uh(z)$ in D are $x = 0$ and, resp., $x = 999999$, if $999999 > th$. These answers correctly indicate low and high uncertainty in D and, resp., D^U.

For computing correct answers to queries (rather than to denials representing constraints), AHI is a viable alternative to CQA. A comparison, which turned out to be advantageous for AHI, has been presented in [10]. Moreover, recall that, with a simple variant of AHI, reasonable answers can be provided even if these answers depend on violated constraints, as seen in Section 5.

7 Related Work

Although integrity and other desirable properties of data of diverse kind are obviously related, it seems that they never have been approached in a uniform way, as in this paper, neither in theory nor in practice.

However, semantic similarities and differences between the integrity of data and data that have or lack some desirable properties are identified in a collection of work on modeling and managing uncertain data [19]. In that book, largely diverse proposals to handle data that lack quality, and hence involve some uncertainty, are discussed. In particular, approaches such as probabilistic and fuzzy set modeling, exception handling, repairing and paraconsistent reasoning are discussed. However, no particular approach to integrity checking is considered.

Several paraconsistent logics that tolerate inconsistency of data have been proposed, e.g., in [3,5]. Each of them departs from classical first-order logic, by adopting some annotated, probabilistic, modal or multivalued logic, or by replacing standard axioms and inference rules with non-standard axiomatizations.

As opposed to that, inconsistency-tolerant integrity checking fully conforms with standard datalog and does not need any extension of classical logic.

Work concerned with semantic inconsistencies in databases is also going on in the field of measuring inconsistency [17]. However, the violation measures on which ITIC is based (cf. Definition 1) have been conceived to work well also in databases with non-monotonic negation, whereas the inconsistency measures in the literature do not deal with non-monotonicity, as argued in [11].

8 Conclusion

In previous papers, we have elaborated novel theories for integrity maintenance [11] (i.e., integrity checking [14] and repairing [12]), as well as for query answering with integrity [10]. Their novelty consisted in a generalization of conventional concepts in terms of inconsistency tolerance, and in using measures to size, compare and control the amount of inconsistency in given database states.

In this paper, we have shown that measure-based methods are able to curtail not only logical inconsistencies, but, more generally, violations of any desirable property that is expressible in the syntax of integrity constraints. This is not possible with conventional approaches that insist, for integrity checking, on the total satisfaction of all constraints in each committed state, and, for reparing, on the total elimination of all constraint violations. As illustrated in Section 6, the use of inconsistency-tolerant approaches is indeed essential, since wrong, possibly fatal conclusions can be inferred from deficient data by methods that are not inconsistency-tolerant.

Additionally, we have shown that information systems can provide valid answers in the presence of violations of desirable properties by a technique called "answers that have integrity" [10]. Moreover, we have taken first steps toward a generalization of AHI, in order to provide reasonable answers that may not perfectly comply with desirable properties.

Apart from the research mentioned in Section 5, ongoing work deals with scaling up the results of this paper to maintaining desirable properties across concurrent transactions in distributed and replicated information systems.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Arenas, M., Bertossi, L., Chomicki, J.: Consistent query answers in inconsistent databases. In: 18th Symposium on Principles of Database Systems, pp. 68–79. ACM Press (1999)
3. Bertossi, L., Hunter, A., Schaub, T. (eds.): Inconsistency Tolerance. LNCS, vol. 3300. Springer, Heidelberg (2005)
4. Bowen, K., Kowalski, R.: Amalgamating Language and Metalanguage. In: Clark, K., Tärnlund, S.-A. (eds.) Logic Programming, pp. 153–172. Academic Press (1982)
5. Carnielli, W., Coniglio, M., D'Ottaviano, I. (eds.): The Many Sides of Logic. Studies in Logic, vol. 21. College Publications, London (2009)

6. Ceri, S., Cochrane, R., Widom, J.: Practical Applications of Triggers and Constraints: Success and Lingering Issues. In: 26th VLDB, pp. 254–262. Morgan Kaufmann (2000)
7. Chakravarthy, U., Grant, J., Minker, J.: Logic-based approach to semantic query optimization. Transactions on Database Systems 15(2), 162–207 (1990)
8. Clark, K.: Negation as Failure. In: Gallaire, H., Minker, J. (eds.) Logic and Data Bases, pp. 293–322. Plenum Press (1978)
9. Decker, H.: The Range Form of Databases and Queries or: How to Avoid Floundering. In: 5th ÖGAI. Informatik-Fachberichte., vol. 208, pp. 114–123. Springer, Heidelberg (1989)
10. Decker, H.: Answers that Have Integrity. In: Schewe, K.-D. (ed.) SDKB 2010. LNCS, vol. 6834, pp. 54–72. Springer, Heidelberg (2011)
11. Decker, H.: Measure-based Inconsistency-tolerant Maintenance of Database Integrity. In: Schewe, K.-D., Thalheim, B. (eds.) SDKB 2013. LNCS, vol. 7693, pp. 149–173. Springer, Heidelberg (2013)
12. Decker, H.: Partial Repairs that Tolerate Inconsistency. In: Eder, J., Bielikova, M., Tjoa, A.M. (eds.) ADBIS 2011. LNCS, vol. 6909, pp. 389–400. Springer, Heidelberg (2011)
13. Decker, H., Martinenghi, D.: Modeling, Measuring and Monitoring the Quality of Information. In: Heuser, C.A., Pernul, G. (eds.) ER 2009. LNCS, vol. 5833, pp. 212–221. Springer, Heidelberg (2009)
14. Decker, H., Martinenghi, D.: Inconsistency-tolerant Integrity Checking. TKDE 23(2), 218–234 (2011)
15. Fung, T.H., Kowalski, R.: The IFF proof procedure for abductive logic programming. J. Logic Programming 33(2), 151–165 (1997)
16. Gupta, A., Sagiv, Y., Ullman, J., Widom, J.: Constraint checking with partial information. In: 13th Symposium on Principles of Database Systems, pp. 45–55. ACM Press (1994)
17. Hunter, A., Konieczny, S.: Approaches to measuring inconsistent information. In: Bertossi, L., Hunter, A., Schaub, T. (eds.) Inconsistency Tolerance. LNCS, vol. 3300, pp. 191–236. Springer, Heidelberg (2005)
18. Lee, S.Y., Ling, T.W.: Further improvements on integrity constraint checking for stratifiable deductive databases. In: 22th International Conference on Very Large Data Bases, pp. 495–505. Morgan Kaufmann (1996)
19. Motro, A., Smets, P.: Uncertainty Management in Information Systems: From Needs to Solutions. Kluwer (1996)
20. Schewe, K.D., Thalheim, B.: Limitations of Rule Triggering Systems for Integrity Maintenance in the Context of Transition Specifications. Acta Cybernetica 13(3), 277–304 (1998)

Heuristics-Based Workload Analysis for Relational DBMSs

Andreas Lübcke, Veit Köppen, and Gunter Saake

University of Magdeburg, Magdeburg, Germany
{andreas.luebcke,veit.koeppen,gunter.saake}@ovgu.de
http://wwwiti.cs.uni-magdeburg.de/iti_db/index_en.php

Abstract. Database systems are widely used in heterogeneous applications. However, it is difficult to decide which database management system meets requirements of a certain application best. This observation is even more true for scientific and statistical data management, because new application and research fields are often first observed in this domain. New requirements are often implied to data management while discovering unknown research and applications areas. That is, heuristics and tools do not exist to select an optimal database management system. We develop a decision support framework to support application-performance analyses on database management systems. We use mappings and merge workload information to patterns. We present heuristics for performance estimation to select the optimal database management system for a given workload. We show, these heuristics improve our decision framework by complexity reduction without loss of accuracy. Finally, we evaluate our heuristics considering standard database benchmarks.

Keywords: Architecture, performance estimation, heuristics, design.

1 Introduction

Database systems (DBS) are pervasively used for all business domains. Therefore, DBS have to manage a huge amount of different requirements for heterogeneous application domains. New data management approaches were developed (e.g., NoSQL-DBMSs [CDG+06, DHJ+07], MapReduce [DG04, DG08], Cloud Computing [AFG+09, FZRL09, BYV08], etc.) to make the growing amount of data[1] manageable for new application domains. We argue, these approaches are solutions that are developed for special applications or need a high degree of expert knowledge. In this paper, we restrict ourselves to relational database management systems (DBMSs). Relational DBMSs are commonly used DBS for highly diverse applications and beside that, relational DBMS are well-known to the mainstream of IT-affine people.

Relational DBMSs[2] are developed to manage data of daily business and reduce paper trails of companies (e.g., financial institutions) [ABC+76]. This approach

[1] Consider the data explosion problem [KS97, NK10].

[2] In the following, we use the term DBMS synonymously for relational DBMS.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 25–36, 2013.

dominates the way of data management that we know as online transaction processing (`OLTP`). Nowadays, fast and accurate forecasts for revenues and expenses are not enough. A new application domain evolves that focuses on analyses of data to support business decisions. Codd et al. [CCS93] defined this type of data analysis as online analytical processing (`OLAP`). Consequently, two disjunctive application domains for relational data management exist with different scopes, impacts, and limitations. We describe both in Section 2.

In recent years, business applications demand for solutions that support tasks from both OLTP and OLAP [Fre97, KN11, Pla09, SB08, VMRC04, ZAL08] thus, coarse heuristics for typical OLTP and OLAP applications have become obsolete (e.g., data warehouses without updates always perform best on column-oriented DBMSs). Nevertheless, new approaches also show impacts and limitations (e.g., in-memory-DBMS, focus on real-time, or dimension updates), such that we argue, there is no DBMS that fits for OLTP and OLAP in all application domains. Heuristics and current approaches for physical design and query optimization only consider a single architecture[3] (e.g., design advisor [ZRL+04], self-tuning [CN07] for row-oriented DBMSs or equivalent for column-oriented DBMSs [Idr10, SAB+05]). As a result, the decision for a certain architecture has to be done beforehand whether this decision is optimal or not for a given workload. Consequently, there are no approaches that do both advice physical design spanning different architectures for OLTP, OLAP, and mixed OLTP/OLAP workloads and estimate which architecture is optimal to process a query/-database operation.

We refine existing heuristics for physical design of DBSs spanning different architectures (e.g., heuristics from classical OLTP domain). In this paper, we consider OLTP, OLAP, and mixed application domains for physical design. That is, we present heuristics that propose the usage of row-oriented DBMSs (`row stores`) or column-oriented DBMSs (`column stores`). Furthermore, we present heuristics for query execution or rather for processing (relational) database operations on column and row stores. These heuristics show which query type or database operation performs better on a particular architecture[4] and how single database operations may affect the performance of a query or a workload. We derive our heuristics from experiences in workload analyses with the help of our decision model, presented in [LKS11]. Further, we evaluate our heuristics based on standardized database benchmarks. We give a brief overview to related topics and conclude afterwards.

2 Column or Row Store: Assets and Drawbacks

In previous work, we already present differences between column and row stores according to different subjects [Lüb10, LS10, LKS11]. We summarize our major observations in the following [LSKS10].

[3] We use the term architecture synonymously for storage architecture.
[4] The term architecture refers to row- and column-oriented database architecture.

The major benefits of column stores wrt. row stores are:

- reduced disk space usage due to aggressive compression (e.g., optimal compression per data type),
- more data fits into main memory thus, less data transfer is needed,
- process data without decompression,
- no I/O overhead for aggregation and column operations which are often used in analytical environments,
- easy inter- and intra-query parallelization
- easily adaptable for fast vector operations.

In contrast, row stores have the following advantages over column stores:

- very easy (and fast) update processing; ACID-transactions with high concurrency are supported,
- self-tuning techniques and advisors that make administration and tuning easier are more advanced,
- intuitive processing of SQL (i.e., easier to predict and understand),
- most (IT-affine) people are familiar with row store technology thus, training cost are less,
- efficient support for a wide field of data management tasks (point- and range-queries, full scans for exploration, etc.).

Of course, both architectures also have disadvantages. We name them briefly because they are mostly complementary to the other architecture. Column stores perform less on update operations and concurrent non-read-only data access due to partitioned data thus, on frequent updates and consistency checks tuple reconstructions cause notable cost. In analytical scenarios, row stores read a lot of unnecessary data because operations are often based on single columns or small subsets of them. Additionally, row stores do not reach compression ratios like column stores because for tuples different data types are combined that cannot be compressed on high ratio as columns that have just one data type and in best case a more uniform structure and length. That is, data size is already larger than for column stores which implies more I/O and more main memory consumption.

In line with others [AMH08, ZNB08], we conclude that both architectures have the right to exist in different applications domains. Neither row stores can outperform column stores nor vice versa.

3 Heuristics on Physical Design

Physical design of DBSs is important as long as DBMSs exist. We do not restrict our considerations to a certain architecture. Further, we consider a set of heuristics that give an outline which architecture is suitable for an application.

Some valid rules exist. First, pure OLTP applications perform best on row stores. Second, classic OLAP application with an ETL (extract, transform, and load) process or rare updates are efficient on column stores[5]. In the following, we consider a more exciting question: In which situation does one architecture outperform the other one and when do they perform nearly equivalent?

OLTP. For OLTP workloads, we just can recommend to use row stores for efficient ACID-support. Tuple reconstructions on updates and data partitioning on inserts decrease significantly the performance of column stores in this domain. A column store does not achieve competitive performance except column-store architecture changes significantly to support concurrent update processing (with ACID-support) efficiently.

OLAP. In this domain, one might assume a similar situation as for OLTP workloads, which is not true in general. We are aware that column stores outperform row stores for many applications or queries; that is, for aggregates and access as well as for processing of a small number of columns, column stores perform better. In the majority of cases, column stores are suitable for applications in this domain. Nevertheless, there exist complex OLAP queries where column stores loose their advantages (cf. Section 2). Complex OLAP queries consecutive groupings, a large number of predicate selections on different columns or complex joins. For these complex queries, row stores can achieve competitive results although they consume more main memory, disk space, etc. We argue, the amount of tuple reconstruction within these queries decreases column-stores performance. Consequently, these queries have to be considered for an architecture selection because they can influence the physical design estimation critically even more if there are more than a few of these queries in workload.

OLTP/OLAP. In these scenarios, your physical design strongly depends on the ratio between updates, point queries, and analytical queries. Our experience is that column stores perform about 100-times slower on OLTP-transactions (updates, inserts, etc.) than row stores. In practice, this observation is more crucial because it does not even consider concurrency (e.g., ACID); that is, our observation bases on single-user execution of transactions. Assuming transaction and analytical queries in mean take the same time, we state that one transaction only occurs every 100 queries (OLAP). The fact that analytical queries last longer than a single iteration leads us to a smaller ratio. Our experience is that ten executions of analytical queries on a column store are of higher advantage than the extra costs by a single transaction. If we have a smaller ration than 10:1 (analyses/Tx) then we can not give a clear statement. We recommend to use a row store in this situation or one knows beforehand that the ratio changes to more analytical queries. If the ratio falls under this ratio for a column store and is not a temporary change then a system switch is appropriate. So, in mixed workloads, it is all about ratio analytical queries to transactions. Note, the ratios 100:1 and 10:1 can change depending on OLAP-query types. We describe this in more detail in Section 4.

[5] Note: Not all column stores support update processing, they support just ETL.

CPU & I/O. For physical design, we state that CPU and I/O load have to be considered. We observe that in average, row stores consume more I/O bandwidth as well as in peak because data sizes are larger a priori. Due to tuple reconstruction and decompression of data, column stores consume more CPU time because reconstruction and decompression are additional computational cost. That is, we have to consider wherever we have reserves in hardware resources.

Our heuristics can be used as guideline for architecture decisions for certain applications. That is, we select the most suitable architecture for an application and afterwards use existing approaches (like IBM's advisor [ZRL+04]) to tune physical design of a certain architecture. If workload and DBSs are available for analysis, we emphasize to use our decision model [LKS11] to compute the optimal architecture for a workload. The presented heuristics for physical design extent our decision model to reduce computation cost (i.e., solution space is pruned). Additionally, heuristics make our decision model available for scenarios where no or only less information is available.

4 Heuristics on Query Execution

In the following, we present heuristics for query execution on complex or hybrid DBS. That is, we assume a DBS that supports column- and row-store functionality or a complex environment with at least two DBMSs containing data redundant whereby at least one DBMS is setup for each OLTP and OLAP. In the following, we only discuss query processing heuristics for OLAP and OLTP/OLAP workloads due to our assumptions above and the fact that there is no competitive alternative to row stores for OLTP.

Listing 1. TPC-H query Q6

```
select   sum(l_extendedprice * l_discount) as revenue
from     lineitem
where    l_shipdate >= date '1994-01-01'
         and l_shipdate < date '1994-01-01' + interval '1' year
         and l_discount between .06 - 0.01 and .06 + 0.01
         and l_quantity < 24;
```

OLAP. In this domain, we face a huge amount of data that generally is not frequently updated. Column stores are able to significantly reduce the amount of data due to aggressive compression. That is, more data can be loaded in main memory and overall I/O between storage and main memory is reduced. We state that this I/O reduction is the major benefit of column stores. We have the experience that row stores perform worse on many OLAP queries because row stores loose performance due to fact that CPUs idle most time while waiting for I/O from storage. Moreover, row stores read data that is not needed for most OLAP queries. In our example from TPC-H benchmark [Tra10], one can see that only a few columns of the lineitem relation have to be accessed (cf. Listing 1). In contrast to column stores, row stores have to access the whole relation (which as fact table is quite large) to answer this query. We state, most OLAP queries fit into this pattern. We recommend using column-store functionality to answer OLAP queries as long as they only access a minority of columns from relation for aggregation and predicate selection.

Complex Queries. OLAP analyses become more and more complex. Thus, queries become more complex, too. These queries describe complex issues and produce large reports. Our example (Listing 2) from the TPC-H benchmark could be part of a report (or a more complex query). Complex queries access and aggregate a major part of relations, that is, nearly the whole relation has to be read. This implies a number of tuple reconstructions that reduce performance of column stores significantly. Hence, row stores achieve competitive performance because nearly the whole relation has to be accessed. Our example shows another reason for a number of tuple reconstructions: group operations. Tuples have to be reconstructed before aggregating groups. Further reasons for significant performance reduction by tuple reconstructions are a large number of predicate selections on different columns as well as complex joins. We argue, this complex OLAP query type can be executed on both architectures. We conclude, we can use for this query type our decision model for load balancing in complex environments and hybrid DBMSs.

Listing 2. TPC-H query Q13

```
select c_count, count(*) as custdist from (
        select c_custkey, count(o_orderkey) from
                customer left outer join orders on c_custkey = o_custkey
                and o_comment not like '%special%request%'
                group by c_custkey) as c_orders (c_custkey, c_count)
group by c_count
order by custdist desc, c_count desc;
```

Mixed Workloads. In mixed workload environments, our first recommendation is to split workload into two parts OLTP and OLAP. Both parts can be allocated to the corresponding part of the DBS with row- or column-store functionality. With this approach, we achieve competitive performance for both OLAP and OLTP because according to our assumption complex systems have to have at least one DBMS of each architecture. A more fine-grained splitting for OLAP-workload parts is not recommended in general. Although row stores can reach competitive performance results on OLAP queries, they do not outperform column stores. Nevertheless, our split methodology can also be used for load balancing if a part of the DBS is busy. Further, we state that a future hybrid system has to satisfy both architectures. Processing mixed workloads with our split behavior has two additional advantages. First, we can reduce issues with complex OLAP queries by correct allocation or dividing over at least two parts. Second, we consider time-bound parameters for queries.

As mentioned above, two integration methods for the query-processing heuristics are available. First, we propose the integration into a hybrid DBMS[6] to

[6] This to the best of our knowledge does not exist so far.

decide where to optimally execute a query. That is, heuristics enable rule-based query optimization for hybrid DBMSs as we know from row-store optimizer [GD87]. Second, we propose a global manager on top of complex environments to optimally distribute queries (as we know from distributed DBMSs [ÖV11]). Our decision model analyzes queries and decides where to distribute queries to. Afterwards, queries are locally optimized by the DBMS itself. Note, time-bound requirements can change over time and system design determines how up-to-date data is in OLAP systems (e.g., real-time load [SB08] is available or not). We assume, such time-bound requirements are passed as parameter to the decision model. That is, even OLAP queries can be distributed to OLTP part of the complex environment if in OLAP part data is insufficiently updated or vice versa, we have to ensure analysis on most up-to-date data. Additionally, such parameter can be used to alternatively allocate high-priority queries to another DBS part if the query has to wait otherwise. Further, this approach is extendable for load balancing methodologies in complex environments. We use monitoring functionality of DBMSs to observe whether one part of the environment is overloaded. We argue, two complementary approaches are suitable for load balancing in complex environments. First, we consider the overall load in the environment and process a query on the less loaded part. Second, we consider our heuristics (cf. Section 3) to eliminate bottlenecks for CPU and I/O. However, we take the load from overloaded parts. Thus, we state that we achieve performance benefits wrt. the overall throughput.

5 Evaluation

In the following, we evaluate our previous discussion. We use the standardized benchmarks TPC-H [Tra10] and TPC-C [Fer06][7] (both with 10GB of data) to show significance of our results. For the performance measurements we use a Dell Optiplex 980[8] and run our tests on Oracle 11gR2.1 (row store) and SybaseIQ 15.2 (column store) with each 1GB main memory available whereby we take measurements every 0.25 seconds. Due to space constraints, we cannot present all results here. We only present a representative selection.

First, we present the execution time for TPC-H on both systems in Table 1. We highlight queries we discussed in Section 4. We consider complex versus classical OLAP query. We can easily see without considering additional values that row store architecture cannot compete with column store architecture for Q6, but for Q13. We identify more examples where the row store achieves competitive result: Q11, Q14, Q17, and Q19. As we assume, for most queries row stores cannot compete to column stores (cf. Table 1); e.g., Q10, Q15). For Q6, this is a classical OLAP query with minor number of columns to access and process. Sybase has an average CPU utilization of 274% and 14.19MB/s (in 1 sec. 0 to 41.69MB/s) average disk read for Q6 whereby CPUs already start work (67%) after first measurement interval (0.25 sec). Oracle's average CPU utilization is approx.

[7] A prepared TPC-C environment; Referring: `http://www.tpc.org`

[8] QuadCore @3.33GHz, 8GB RAM running an Ubuntu 10.04LTS (2.6.32-41).

Table 1. Execution time of TPC-H queries (in mm:ss)

Query	Oracle	Sybase	Query	Oracle	Sybase
Q1	01:40	00:58	Q12	01:41	01:26
Q2	01:21	00:28	**Q13**	**00:52**	**00:42**
Q3	01:52	00:47	Q14	01:24	02:16
Q4	01:38	00:35	Q15	01:22	00:20
Q5	03:03	00:25	Q16	00:09	00:07
Q6	**01:21**	**00:05**	Q17	01:22	01:14
Q7	02:12	00:06	Q18	03:51	01:05
Q8	01:47	00:21	Q19	01:23	02:05
Q9	03:42	02:30	Q20	01:33	00:50
Q10	02:00	00:15	Q21	04:08	02:22
Q11	00:13	00:10	Q22	00:20	00:11

85%, but it needs more than one second to start data processing, and approx. 95MB/s (over 01:19min) average disk read. These values show how different is the amount of data to process for both systems. For this type of query, it is reasonable that row stores cannot achieve competitive results. We can confirm these results for Q10 (Sybase: approx. 22MB/s and 220%, Oracle: approx. 92MB/s (read) 9MB/s (write during dump, 89% (12% during dump)) and Q15 (Sybase: approx. 23MB/s and 125%, Oracle: approx. 98MB/s and 87%) as well. Even worse, Q10 (buffering) has a long write phase in Oracle. Sybase only writes the final result. This behavior corresponds to our heuristics before; Oracle uses the whole I/O bandwidth, but cannot utilize all available CPU performance due to the large amount of data. Moreover, main memory is not sufficient for computations. For Q13, Sybase cannot use parallel processing on different columns due to the grouping operations that need tuple reconstructions. We can observe this fact by the CPU usage (approx. 100%) that shows Sybase mainly uses 1 CPU in contrast to other queries. The disk usage does not significantly change (approx. 14MB/s). Oracle does not need the whole I/O bandwidth (approx. 34MB/s) and does not swap to disk. Even CPU usage stagnates at approx. 80%. This behavior can be explained by the iterative execution of Q13, that is the inner grouping has to be computed at first. For column stores, this means tuple reconstruction for each grouping; for row stores, it means to process as usual just only one step after the other. Such effect can more frequently occur for complex OLAP queries. In Q11, we have the same situation caused by the complex having clause (Sybase: approx. 19MB/s and 94%, Oracle: approx. 70MB/s and 94%) that contains an aggregation over two columns and a sub-query as selection predicate. Q14 is quite similar to the other queries but Q19 is completely different. In Q19 a huge number of predicate selections, that are not independent from each other, causes the behavior for column stores. Column stores have to reconstruct tuple for lineitem relation (fact table) as well as the part relation.

Second, we investigate column-store performance on OLTP workloads. We show that column stores cannot sufficiently support mixed OLTP/OLAP workloads. Consequently, we argue that row-store functionality is needed for hybrid

DBMSs and complex DBS environments. We use the TPC-C benchmark to simulate an OLTP workload. We exclude TPC-C transaction 2.6 because it is just a cursor declaration for in-memory tasks. Thus, we do not see an effect. Results for OLTP on Oracle and Sybase are listed in Table 2. For Transactions 2.4 and 2.5, we can observe relatively good performance for Sybase even if the execution time is many times over Oracle's execution time. We argue, the relatively good performance of Sybase is a result of the ratio between lookup and update/insert operations (approx. 1:1), because in general Sybase shows very good performance on lookups. We can prove the good performance of Sybase on lookups with transaction 2.8 that contains only two select-queries. Both systems achieve nearly equivalent performance. In contrast, Sybase shows poor performance for transaction 2.7, that contains twice as many update operations as lookups. Our results show that column stores are not suitable for OLTP workload even when they have read-write storage like Sybase. The update support of column stores is more a workaround than a replacement of classical OLTP systems.

Table 2. Execution time (in sec.), Ø CPU in %, and Ø I/O (in MB/s) of TPC-C transactions

	Oracle			Sybase		
Tx	Time	CPU	I/O	Time	CPU	I/O
2.4	<1	73	2.99	5	65	18.15
2.5	<1	62	29.35	4	75	12.07
2.7	<1	68	4.65	40	139	9.05
2.8	<1	63	8.20	<1	198	11.51

The TPC-H- and TPC-C-benchmark results confirm our heuristics for CPU and I/O consumption. Sybase shows relatively high I/O on TPC-C because data has to be reread for transaction processing. That is, column stores in general consume more CPU time, whereas row store consume more I/O bandwidth.

6 Related Work

Several approaches have been developed to analyze and classify workloads, e.g., [HGR09, Raa93]. These approaches recommend tuning and design of DBS, try to merge similar tasks, etc. to improve performance of DBSs. At present, workload-analysis approaches are limited to classification of queries to execution pattern or design estimations for a certain architecture; that is, the solution space is pruned before analysis and performance estimations are done. With our decision model and heuristics, we propose an approach that is independent from architectural issues.

Due to success in the analytical domain, researchers devote greater attention on column stores whereby they focus on analysis performance [Aba08, Idr10] on the one hand and on the other hand to overcome update-problems with separate storages [Fre97, Aba08]. However, a hybrid system in an architectural manner does not exist. In-memory-DBMSs are developed in recent years (e.g., Hana [Pla09], HyPer [KN11]) that satisfy requirements for mixed OLTP/OLAP workloads. Nevertheless, we state that even today not all DBSs can run in-memory (due to monetary or environmental constraints). Thus, we propose a more general approach that is also suitable for disk-based DBSs.

7 Conclusion

In recent years, many new approaches as well as new requirements encourage performance of DBMSs in many application domains. Nevertheless, new approaches and requirements also increase complexity of DBMS selection, DBS design, and tuning for a certain application. We focus our considerations on OLTP, OLAP, and OLTP/OLAP workloads. That is, we consider which approach is most suitable for a given workload.

Therefore, we present heuristics on design estimations and query execution for the two relational storage architectures: row and column stores. We use our decision model [LKS11] to observe the performance of several applications. Consequently, we derive heuristics from our experiences while evaluating our decision model. We present heuristics for DBS design to a priori select the most suitable DBMS and tune afterwards. This approach avoids misleading tuning if the architecture selection is wrong. Furthermore, we presented heuristics on query execution for both storage architectures. On the one hand, we want to emphasize our heuristics for physical design. On the other hand, we propose these heuristics for integration in hybrid DBS whether it is a real hybrid DBMS that supports either both architectures or it is a complex system that consists of different DBMSs (at least one row and one column store). Summarizing, there is no alternative in OLTP environments to row stores, for OLAP applications we recommend to use column stores taking into account that a number of complex OLAP queries can change this recommendation, and finally in mixed OLTP/OLAP environments the focus is on the ratio of OLAP queries to OLTP transactions (again taking very complex OLAP queries into account). We show relevance and correctness of our heuristics with an experiment. From our point of view, our heuristics are a first step to rule-based query optimization in hybrid systems/architectures.

In future work, we will evaluate and refine our heuristics in combination with our decision model. After refinement, we will implement the heuristics in our decision model. Thus, we create a design advisor for both relational storage architectures. We will combine the design advisor and decision model to a runtime query-distributor for load balancing in complex environments. Finally, we plan to implement a hybrid DBMSs using our heuristics for ruled-based query optimization.

Acknowledgements. We thank Martin Kuhlemann and Martin Schäler for helpful discussions and comments on earlier drafts of this paper. This work is partially supported by the German Ministry of Education and Research within the ViERforES-Project (grant no. 01IM10002B).

References

[Aba08] Daniel, J.: Abadi. Query execution in column-oriented database systems. PhD thesis, Cambridge, MA, USA, Adviser: Madden, Samuel (2008)

[ABC$^+$76] Astrahan, M.M., Blasgen, M.W., Chamberlin, D.D., Eswaran, K.P., Gray, J., Griffiths, P.P., Frank King III, W., Lorie, R.A., McJones, P.R., Mehl, J.W., Putzolu, G.R., Traiger, I.L., Wade, B.W., Watson, V.: System R: Relational Approach to Database Management. ACM Trans. Database Syst. 1(2), 97–137 (1976)

[AFG$^+$09] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley (February 2009)

[AMH08] Abadi, D.J., Madden, S.R., Hachem, N.: Column-stores vs. row-stores: How different are they really?. In: SIGMOD 2008, pp. 967–980 (2008)

[BYV08] Buyya, R., Yeo, C.S., Venugopal, S.: Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. In: HPCC 2008, pp. 5–13 (2008)

[CCS93] Codd, E.F., Codd, S.B., Salley, C.T.: Providing OLAP to User-Analysts: An IT Mandate. Ann ArborMichigan 24 (1993)

[CDG$^+$06] Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.: Bigtable: A Distributed Storage System for Structured Data. In: OSDI 2006, pp. 205–218 (2006)

[CN07] Chaudhuri, S., Narasayya, V.: Self-tuning database systems: A decade of progress. In: VLDB 2007, pp. 3–14 (2007)

[DG04] Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: OSDI 2004, pp. 137–150 (2004)

[DG08] Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM 51(1), 107–113 (2008)

[DHJ$^+$07] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels, W.: Dynamo: amazon's highly available key-value store. In: SOSP 2007, pp. 205–220 (2007)

[Fer06] Ferraris, D.R.L.: TPCC-UVa: an open-source TPC-C implementation for global performance measurement of computer systems. SIGMOD Record 35(4), 6–15 (2006)

[Fre97] French, C.D.: Teaching an OLTP database kernel advanced datawarehousing techniques. In: ICDE 1997, pp. 194–198 (1997)

[FZRL09] Foster, I.T., Zhao, Y., Raicu, I., Lu, S.: Cloud Computing and Grid Computing 360-Degree Compared. CoRR 2009, abs/0901.0131 (2009)

[GD87] Graefe, G., DeWitt, D.J.: The EXODUS Optimizer Generator. In: SIGMOD 1987, pp. 160–172 (1987)

[HGR09] Holze, M., Gaidies, C., Ritter, N.: Consistent on-line classification of DBS workload events. In: CIKM 2009, pp. 1641–1644 (2009)

[Idr10] Idreos, S.: Database Cracking: Torwards Auto-tuning Database Kernels. PhD thesis (2010)

[KN11] Kemper, A., Neumann, T.: HyPer: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots. In: ICDE 2011, pp. 195–206 (2011)

[KS97] Korth, H.F., Silberschatz, A.: Database Research Faces the Information Explosion. Commun. ACM 40(2), 139–142 (1997)

[LKS11] Lübcke, A., Köppen, V., Saake, G.: A Decision Model to Select the Optimal Storage Architecture for Relational Databases. In: Proceedings of the Fifth IEEE International Conference on Research Challenges in Information Science, RCIS 2011, pp. 74–84 (2011)

[LS10] Lübcke, A., Saake, G.: A Framework for Optimal Selection of a Storage Architecture in RDBMS. In: DB&IS 2010, pp. 65–76 (2010)

[LSKS10] Lübcke, A., Schäler, M., Köppen, V., Saake, G.: Workload-based Heuristics for Evaluation of Physical Database Architectures. In: DB&IS 2012, pp. 3–10 (2012)

[Lüb10] Lübcke, A.: Challenges in Workload Analyses for Column and Row Storess. In: Grundlagen von Datenbanken (2010)

[NK10] Naydenova, I., Kaloyanova, K.: Sparsity Handling and Data Explosion in OLAP Systems. In: MCIS 2010, pp. 62–70 (2010)

[ÖV11] Tamer Özsu, M., Valdurie, P.: Principles of Distributed Database Systems, 3rd edn. Springer (2011)

[Pla09] Plattner, H.: A common database approach for OLTP and OLAP using an in-memory column database. In: SIGMOD 2009, pp. 1–2. ACM (2009)

[Raa93] Raatikainen, K.E.E.: Cluster Analysis and Workload Classification. SIGMETRICS Performance Evaluation Review 20(4), 24–30 (1993)

[SAB+05] Stonebraker, M., Abadi, D.J., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., Lau, E., Lin, A., Madden, S., O'Neil, E.J., O'Neil, P.E., Rasin, A., Tran, N., Zdonik, S.B.: C-Store: A column-oriented DBMS. In: VLDB 2005, pp. 553–564. VLDB Endowment (2005)

[SB08] Santos, R.J., Bernardino, J.: Real-time data warehouse loading methodology. In: IDEAS 2008, pp. 49–58 (2008)

[Tra10] Transaction Processing Performance Council. TPC BENCHMARK™ H. White Paper, Decision Support Standard Specification,Revision 2.11.0 (April 2010)

[VMRC04] Vaisman, A.A., Mendelzon, A.O., Ruaro, W., Cymerman, S.G.: Supporting dimension updates in an OLAP server. Information Systems 29(2), 165–185 (2004)

[ZAL08] Zhu, Y., An, L., Liu, S.: Data Updating and Query in Real-Time Data Warehouse System. In: CSSE 2008, pp. 1295–1297 (2008)

[ZNB08] Zukowski, M., Nes, N., Boncz, P.A.: DSM vs. NSM: CPU performance tradeoffs in block-oriented query processing. In: DaMoN 2008, pp. 47–54 (2008)

[ZRL+04] Zilio, D.C., Rao, J., Lightstone, S., Lohman, G.M., Storm, A.J., Garcia-Arellano, C., Fadden, S.: DB2 Design Advisor: Integrated automatic physical database design. In: VLDB 2004, pp. 1087–1097. VLDB Endowment (2004)

Information Technology for the Decision-Making Process in an Investment Company

Tetiana Zakharova and Valentyna Moskalenko

National Technical University ’Kharkiv Polytechnic Institute’,
Frunze Street 21, 61024 Kharkiv, Ukraine
tetiana.zakharova@gmail.com, mvv_17@mail.ru

Abstract. This paper covers one of the directions of the Investment Company functioning, namely investments in business projects of the national economy. The information technology in order to justify and make investment decisions has been developed. This technology is the complex of models and techniques of analysis of the investment market conditions, formation of the investment policy, evaluation of effectiveness and risks of investment directions and projects, construction of investment portfolio. The implementation of the Decision Support System for the Investment Company is considered. Main functions and requirements of the system are presented. The software solution structure based on service-oriented architecture (SOA) is considered. The description of packages, interfaces and their interaction is described. The developed application is a complete solution for operating with the internal and external data of the Investment Company, obtaining the results of assessment and report generation.

Keywords: modeling, Investment Company, investment policy, investment portfolio, DSS, service-oriented architecture.

1 Introduction

In this paper the decision-making process in an Investment Company is considered. Implementation of the process steps meets a lot of investment challenges taking into account the impact of great number of external environment factors and parameters of the Investment Company internal state on company’s function results (outputs). Heterogeneous information from different sources is used for this purpose. Typically this information refers to the different temporal periods and is characterized by large volumes. These facts are the evidence of necessity of development and deployment of the information technology into the Investment Company management process. This technology should enable to perform data collection, processing and storage functions as well as transformation of input data into new advanced information through mathematical models and algorithms in order to support the investment decision-making process in the company.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 37–48, 2013.

2 The Purpose of the Article

Now most software products for companies which carry out investment activity solve individual investment tasks. For example, estimation and analysis of securities, estimation of projects efficiency are introduced in some software products, in other products options to prepare business plan and to carry out the budgeting process are introduced. Also, there are systems applied in project management such as Microsoft Project, PrimeExpert, Primavera, and other that can be used for investment tasks solving. Software products for the financial management of companies, for individual finance and credit organizations are also available at the information technologies market. However, these products are inapplicable for the multifaceted task solving which would cover all stages of the investment decision-making process.

Based on the analysis of the existing data-analytical systems applied in the investment field it is possible to infer that some of tasks which allow making effective investment decisions are not performed by the offered systems (for example, the task of determination of the investment market situation, the task of formation of the investment policy depending on the type of the forecast situation). Therefore, the Decision Support System (DSS) as realization of the certain information technology which enables solving the complex of investment tasks is considered in this work. This DSS implements: a) models and methods of analysis of the investment market condition, b) algorithm of the investment policy formation considering the type of the forecast situation at the market of investments, c) technique of investment projects efficiency and risk assessment for different type of initial (input) information, d) models and techniques of investment projects portfolio construction within a certain investment policy. Also, this system contains a number of components for information support of the decision-making process.

3 Investment Decision-Making Technique

Based on problem analysis in the field of investments the technique of decision-making in the investment companies was developed as sequence of business-processes in this work. This technique is presented in IDEF0 notation (Fig.1) [1].

Information on the investment projects and on parameters characterizing the state of the external environment is the input data. The results of this sequence of business-processes implementation are the forecast situation at the investment market, models of the investment policies and constructed investment portfolio. Experts, management of Investment Company, investors (namely, their preferences related to the chosen investment policy), and applied software are the mechanisms. Cognitive approach, correlation analysis, technique of evaluation of the future situation at the investment market, technique of investment policy forming, technique of assessment of investment attractiveness of branches of the national economy, technique of efficiency of investment projects estimation, technique of investment projects portfolio construction are the rules and procedures that regulate implementation of these business-processes.

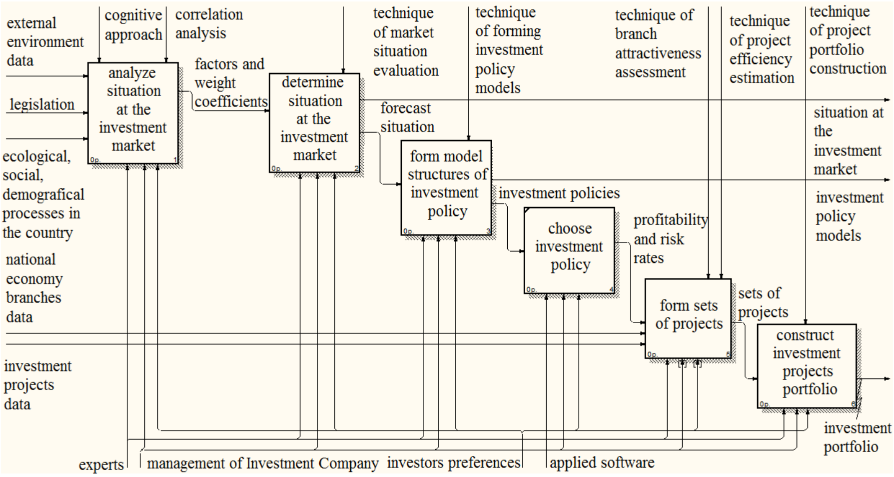


Fig. 1. Technique of Decision-Making in the Investment Company

Stages of the investment decision-making technique are described below.

1. Firstly, a prospective situation at the investment market is analyzed. Namely the analysis of macroeconomic and specific indexes indicating the state of the national economy is carried out. The cognitive map is developed for this purpose. Set of factors $\{x_i,\ i=\overline{1,\widehat{m}}\}$ is selected, where $\widehat{m}$ is the whole number of factors. This map could be presented as an oriented graph where the factors characterizing a situation are tops and the dependences between factors are arcs. It is possible to build such a graph through an adjacency matrix $A=\{a_{ij}\}$ the elements of which specify the presence of arc from the top '*i*' to the top '*j*'. Thus, $a_{ij}=1$ if there is an arc from the top '*i*' to the top '*j*' in the graph, $a_{ij}=0$ if there is no arc from the top '*i*' to the top '*j*' in the graph. Then the most significant factors for description of investment situation are selected. In order to find these significant factors this work suggests to explore factors that influence the situation at the investment market directly and indirectly (through one intermediate factor). Thereby, set of factors is the following $\{x_i,\ i=\overline{1,m}\}$, $m\le\widehat{m}$, where m is the number of significant factors. Then the comparative factor estimation and the exclusion of the correlated factors are carried out. It could be achieved through the correlation coefficients analysis. For this purpose correlation coefficients measuring the closeness degree between all pairs of factors are estimated.

$$r_{x_i x_j} = \frac{\sum_{h=1}^{H}(\tilde{x}_{i,h} - \bar{x}_i)(\tilde{x}_{j,h} - \bar{x}_j)}{\sqrt{\sum_{h=1}^{H}(\tilde{x}_{i,h} - \bar{x}_i)^2} \cdot \sqrt{\sum_{h=1}^{H}(\tilde{x}_{j,h} - \bar{x}_j)^2}}, \tag{1}$$

where x_i and x_j – compared factors, $i, j = \overline{1,m}$, $\tilde{x}_{i,h}$ and $\tilde{x}_{j,h}$ - x_i and x_j factors values at the moment h, $\bar{x}_i$ and $\bar{x}_j$ - factors mean values, H - amount of sampling for each factor. The insignificance of correlation coefficient $r_{x_i x_j}$ is tested by Student's *t*-criterion. If the correlation coefficient takes value from $[\xi_1; 1]$ or $[-1; \xi_2]$ then the factors are correlated and one of them should be excluded. It is recommended to assign such values for the parameters $\xi_1 = 0{,}7$ and $\xi_2 = -0{,}7$ in the work.

The set of factors ranked by the degree of importance is formed at the next stage. Experts assign a rank against every factor in questionnaires in accordance to the rank scale. This scale measures the factor impact on an investment situation. Thus, the most significant factor has a rank '1', less significant – rank '2', etc. The experts are selected in such way that their competences are considered to be equivalent. In the case of different experts' experience, the competence coefficients are to be assigned to them. The results of questioning are filled out in a table called a matrix of ranks. The dimension of this matrix is $K \times \tilde{m}$, where *K* is the amount of experts, $\tilde{m}$ is the amount of independent factors. Conformity of experts' opinions is checked up through the coefficient of concordance. All factors are ordered as far as the decrease of their influence the investment situation by two methods – middle rank and medians of ranks. Thus, the ranked set of independent factors which substantially influence the functioning of the investment market is $\{x_i, i = \overline{1,\tilde{m}}\}$, $\tilde{m} \le m$ with the final rank values $\tilde{\eta}_i$. The weight coefficient is determined for every factor:

$$\rho_i = \frac{\tilde{\eta}_i}{\sum_{i=1}^{\tilde{m}} \tilde{\eta}_i}. \tag{2}$$

2. The ranked set of factors and their weight coefficients are the basis for forecasting the parameters that characterize the future situation at the investment market. Different forecasting methods for parameters that characterize the most significant factors are to be applied. The method which gives the minimum error (and/or greater stability and adequacy) of forecasting will be chosen. In other words, it is very important to take into account the error of forecasting and to minimize it due to application of adequate methods for significant factors. Such procedure of

forecasting methods choice could not be used for less important factors because it requires more time and other resources.

Because all parameters that describe the impact of factors on an investment situation have a different dimensions, it is offered to put some value of the aggregated index in accordance to the certain type of investment situation. It is necessary to put some score in accordance with the every forecasting parameter value.

Indexes parameter values are transformed to the dimensionless and equal domain of variation. Monotone transformation of parameter values to the normalized dimensionless for all factors is made. Thus, it is necessary to carry out the following transformation for all factors:

$$\omega_i(x_i) = \begin{cases} \dfrac{\theta_i^{\min} - x_i}{\theta_i^{\min} - \theta_i^{\max}}, i \in \tilde{M}^1 \\ \dfrac{\theta_i^{\max} - x_i}{\theta_i^{\max} - \theta_i^{\min}}, i \in \tilde{M}^2 \end{cases}, \tag{3}$$

where $\theta_i^{\min}$ - is a minimum parameter value of factor *i*, $\theta_i^{\max}$ - is a maximum parameter value of factor *i*, $\tilde{M}^1$ - factors which growth positively influences the situation at the investment market, $\tilde{M}^2$ - factors which growth negatively influences the situation at the investment market, $\tilde{M} = \tilde{M}^1 \bigcup \tilde{M}^2$, where $\tilde{m} = \left|\tilde{M}\right|$, $\tilde{m}^1 = \left|\tilde{M}^1\right|$, $\tilde{m}^2 = \left|\tilde{M}^2\right|$, and $\tilde{m} = \tilde{m}^1 + \tilde{m}^2$. It being known that x_i is equal to:

$$x_i = \begin{cases} \theta_i^{\max}, \text{ if } \theta_i^f \geq \theta_i^{\max} \\ \theta_i^{\min}, \text{ if } \theta_i^f \leq \theta_i^{\min} \\ \theta_i^f, \text{ if } \theta_i^{\min} < \theta_i^f < \theta_i^{\max} \end{cases}, \tag{4}$$

where θ_i^f - is a forecast parameter value of factor *i*.

Having studied statistics in Ukraine during different periods experts determine the range of parameter value changes. Also forecast parameter values of factors are transformed to the dimensionless view ($\omega_i(x_i)$). Then the aggregated index that identifies the certain type of situation at the investment market is equal to:

$$S_{MS} = \sum_{i=1}^{\tilde{m}} \rho_i \cdot \omega_i(x_i). \tag{5}$$

Depending on S_{MS} value the type of the situation at the investment market is determined in accordance with the accepted scale:

crisis: $S_{MS} = [0; \delta_1)$;

slight growth: $S_{MS} = [\delta_1; \delta_2)$;

moderate growth: $S_{MS} = [\delta_2; \delta_3)$;

intensive growth: $S_{MS} = [\delta_3; 1]$.

Values δ_i, $i = \overline{1,3}$ are assigned by experts based on the analysis of the national economy state.

3. Then the set of investment policy model structures that corresponds to the situation at the investment market is formed (fig.2). As the investment policy represents the relation between risk and profitability, it is suggested to find the range of weight coefficients of these criteria in this work:

$$\overline{\rho_1}, \overline{\rho_2} \geq 0; \sum_{i=1}^{2} \rho_i = 1. \tag{6}$$

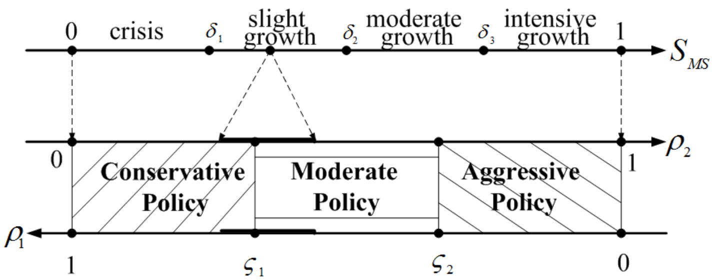


Fig. 2. The 'Market Situation-Investment Policy' relation

To specify the investment policy either trade-off decision or convolution of two criteria could be used to determine weight coefficients of risk and profitability.

Due to the fact that the process of policy formation is related to the analysis of many factors and forecasting of great number of market situation parameters, it is offered to carry out the choice of the policy type based on the Analytic Hierarchy Process (AHP) also. This method enables to consider influence of different factors, subjects of investment market, objectives of an Investment Company, etc., on the Investment Company policy; to carry out the decomposition of choice problem into components and link them in a hierarchy. In other words, with the help of AHP not only policy choice could be carried out but also its forming, i.e. including in the policy other indexes than 'profitability/risk' ratio as well as objectives of the Investment Company, descriptions of the investment market, etc. [2].

4. The next stage is the choice of the investment policy. The choice is carried out on the basis of situation at the investment market, set of structures of investment policy models and investor preferences.

5. At this stage projects-applicants for a portfolio construction are selected. The selection of projects can be carried out on the basis of acceptable level of risk and profitability. An additional criterion can be a particular branch affiliation of the project. This additional option requires assessment of investment attractiveness of branches, and next the estimation of investment projects by the profitability and risk criteria. This stage includes the estimation of the investment projects based on different efficiency criteria (Net Present Value, Profitability Index, Payback Period, etc.) and risk criteria (semideviation of Net Present Value, degree of nonreceipt of the expected income and profitability of all projects).

6. The final stage of the investment decision-making is the project portfolio construction. Thus, it is possible to construct different types of portfolios depending on the aims of the company or the investor (the client of Investment Company). The task of portfolio construction in a general view could be represented as a task of the boolean programming: to find such a vector $\{s_i\}$ that provides an optimum to the criterion of total efficiency of all projects:

$$\mathrm{Pr}\,ofitability = \sum_{i=1}^{p} P_i \cdot s_i \rightarrow \max, \tag{7}$$

and optimum to the criterion of total portfolio risk:

$$Risk = R \rightarrow \min, \tag{8}$$

within the investment resources limits:

$$\sum_{i=1}^{p} I_i \cdot s_i \leq Q, \tag{9}$$

where $s_i = \{0, 1\}$ - boolean variable, $s_i = 1$ if a project is included into portfolio, and $s_i = 0$ if a project is not included into portfolio; P_i - efficiency of the project *i* (for example, *NPV* or *PI* criterion), $i = \overline{1, p}$, where p – number of projects; R - total portfolio risk (it could be semideviation of *NPV* or degree of nonreceipt of the expected income and profitability of all projects); I_i - investment resources of project *i*, $i = \overline{1, p}$; *Q* – total investment resources.

This is a multicriteria optimization task. The choice of method of its solving depends on the formed investment policy, i.e. on risk and profitability ratio. For example, if conservative policy is chosen, the task (7)-(9) will be simplified to one-criterion optimization task with a criterion (8), and a criterion (7) will be transformed into constraint.

Possibility of both static and dynamic projects portfolio construction is considered in this work [3].

Static portfolio is formed at the certain moment of time under the condition that the terms of start and realization coincide for all projects or that the main profit will be received within the considered forecasting time-frame (planning horizon). Within this time the investor policy cannot be changed. Such portfolio can be formed for an external investor (Investment Company client) in the case when the company's activity is the trust assets management.

The dynamic portfolio is a sequence of static portfolios formed according to strategic period intervals. This portfolio will be formed with regard to the investment policy changes. First, the investment policies as the level of risk and profitability and limit of initial investments are formed for each interval of the strategic period. Then the projects for each interval are selected. After that the iterative process of portfolio construction begins. The optimal portfolio using the profitability criterion is constructed in the first interval. All the projects which were not included in this portfolio could be shifted to the next interval in case a client agrees to get the investments in the next interval. The client can decline the investments. In this case the projects that have not been included in the interval can substitute the projects declined by the client. The iterative process is finished when the chosen investment policy is implemented.

Since the dynamic portfolio is formed under the condition that the Investment Company is a direct investor, the Investment Company management will be able to plan cash flows, to increase the level of profitability of the laid-down capital, to improve the financial indexes of the company, to extend possibilities of investing, and, eventually, to revise the strategy of the company development.

4 Decision-Making Information Technology Implementation

On the first stage of design of the software solution that implements the Investment Decisions Support System (IDSS) [4] the main system functions and requirements to it are formalized as a use-case diagram in UML notation (fig. 3).

While designing the solution implementing the IDSS in the Investment Company the concept described in works [5, 6] is used. The functional structure of this solution is presented on the fig. 4 as the diagram of packages in UML notation.

Three main packages such as 'Policy Formation', 'Portfolio Construction' and 'Database' are spotted out. They implement business-logic corresponding to the name of a package. The diagram illustrates both interaction between packages and interaction between package elements.

Software solution interfaces are implemented based on the ASP.NET technology (.NET Framework platform). The purpose of packages and interfaces is the following.

1. The experts interface is used as an incoming interface for macroeconomic indexes, information about branches of national economy and about indexes indicating a situation at the investment market as well as incoming data of projects taken from business plans and expert estimations. This data is added to the database (DB) 'Initial information' by an operator or a decision-former (DF), and also by experts working in the investment field (for example, experts of the Ukrainian Association of Investment Business, Association of Private Investors, Professional Association of Corporate Management etc.), and by experts of the company.

2. The interface of DF is used as an incoming interface for actualization of the initial data; for solving such tasks as the investment policy choice and the investment portfolio construction; for the choice of task solving algorithms and for generating the reports. It has wide possibilities for visualization and analysis of the obtained results.

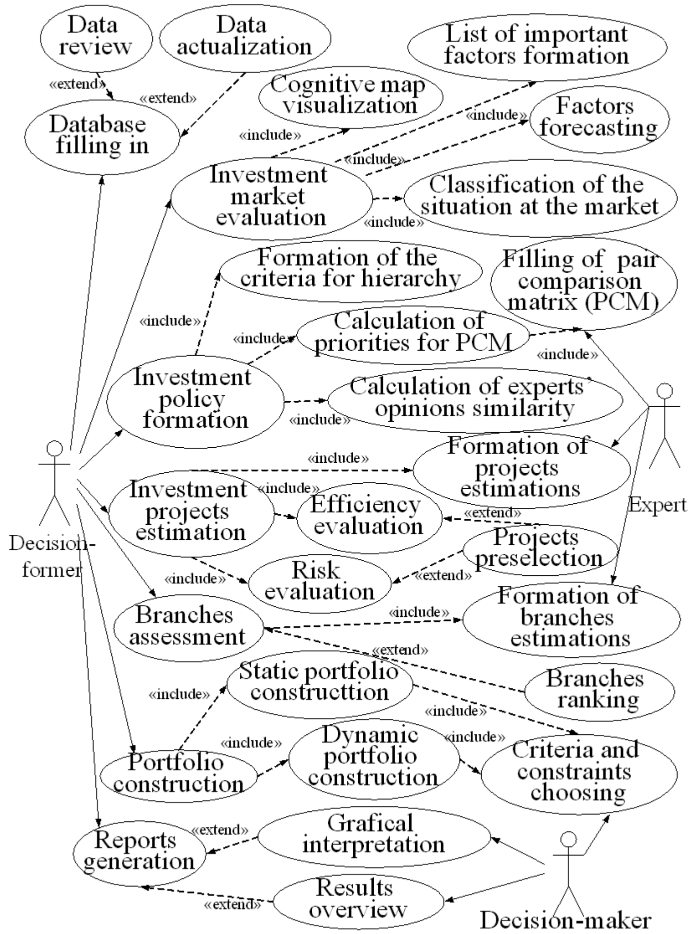


Fig. 3. Use-case diagram of DSS

3. The decision-maker (DM) interface is used as an incoming interface for priority criteria choice while forming the investment policy and for analysis of results of the solved investment tasks made by DF. The investment manager or other person forming an investment policy can be DM.

4. Package 'Policy Formation' is intended for implementation of situation type estimation technique at the investment market as well as implementation of AHP for investment policy formation. All necessary data is in DB 'Company information' and in DB 'Initial information' where data for forecasting of indexes characterizing a situation at the investment market, data for forming of weight coefficients of these indexes, and also expert information concerning implementation of AHP and other necessary information are stored.

5. Package 'Portfolio Construction' is intended for implementation of models and techniques of investment portfolio construction as well as algorithms of investment projects estimation and investment attractiveness of branches of the national economy

estimation. All necessary data is in DB ‘Company information’ and DB ‘Initial information’ where data on investment projects, information on indexes characterizing the state of branches of the national economy are stored. Both static and dynamic portfolio construction algorithms are realized in this package.

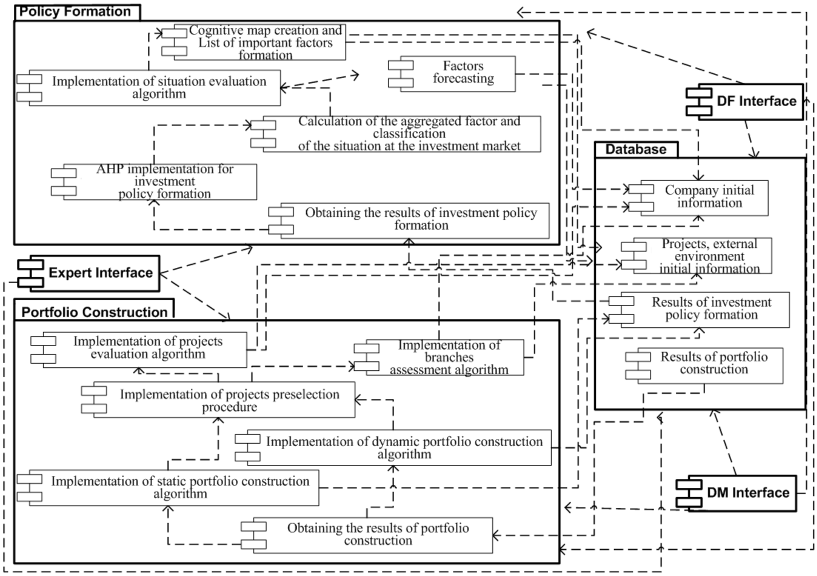


Fig. 4. Functional structure of information technology

6. Package ‘Database’ provides users and applied functional subsystems of IDSS with information resource during all stages of making decisions. It is represented by ‘Company information’ and ‘Initial information’ databases which are implemented in DBMS of MS SQL Server.

The IDEF1X model of ‘Initial information’ database has been elaborated in this work [7]. It is presented on figure 5.

On fig. 6 interaction between packages and components is illustrated.

Service-Oriented Architecture (SOA) is used as a software solution for IDSS. It implies the module approach to software development based on the usage of services with the standardized interfaces. This approach results from the fact that SOA lends itself to implementation of large corporate applications. This software solution is performed as a number of web-services integrated through SOAP protocol that simplifies integration, interaction and distribution of this solution in the current company management system where this IDSS is being deployed.

Packages ‘Policy Formation’ and ‘Portfolio Construction’ are implemented as separate services. Due to the nature of these packages they could be considered and used as individual tasks. Interaction between services, requests to services and responses from services, are realized as SOAP protocol through HTTP. Actually such an approach enables documents exchanging in XML format. Every service contains WSDL (Web Services Description Language) as a description of external interfaces based on XML and components for interaction with other package. Every package contains several components (fig. 4) which are implemented as a number of classes in order to realize certain algorithms. Interaction between components in the framework of package is carried out within interfaces.

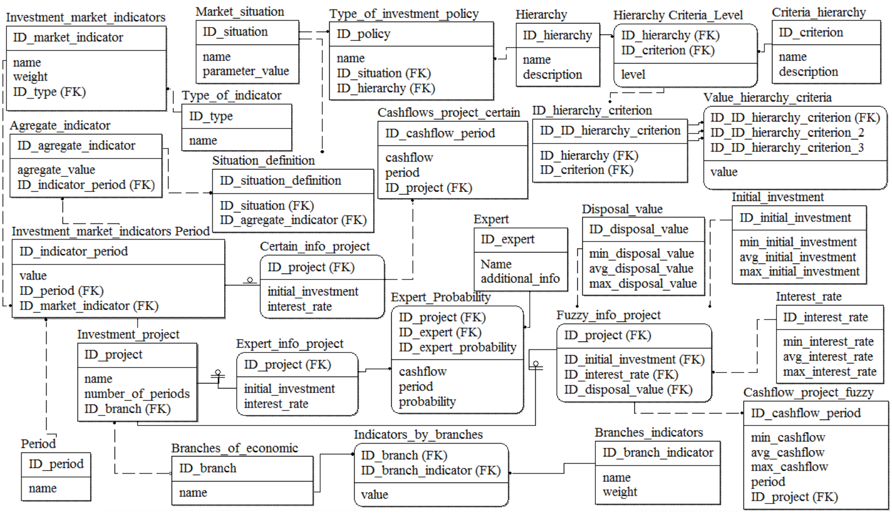


Fig. 5. IDEF1X model for ‘Initial information’ database

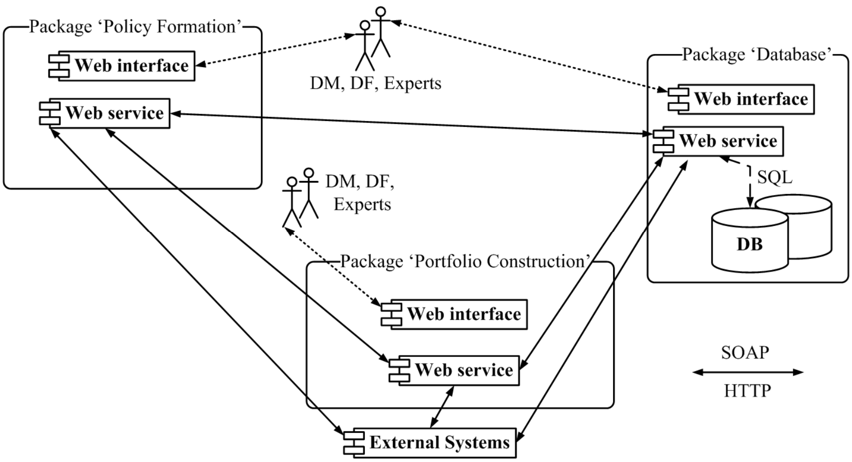


Fig. 6. Interaction between services

DM, DF and Experts use Web Interface to interact with the system. It is possible to upload required information into the service using Web Interface. After it, the request to the Web Service(s) is sent. Information from the Database could be required during the request processing - for this purpose ‘Database’ package contains Web Service. Once this information is processed the result will be sent to the Web Interface or other

Web Service(s). Based on this approach a user can use one or several packages: ‘Policy Formation’, ‘Portfolio Construction’ and ‘Database’. As a result of this approach the system can be extended with an additional Web Service(s) to enable business agility.

The application of this information technology has forecast the slight growth of the national economy. Proceeding from this fact the moderate investment policy was recommended. 19 investment projects from different branches have been explored and the dynamic portfolio has been constructed. As a result, the profitability of company activity has increased by 7 percent.

5 Conclusions

Thus, information technology for decision-making support in the Investment Company has been described in this work. The developed technology enables to make effective investment decisions. This is achieved by solving the investment tasks comprehensively, that is including the analysis and forecasting of the investment market situation, formation of the investment policy, investment projects estimation, and investment projects portfolio construction in the framework of the chosen investment policy. This software solution can be used either separate one within the investment management or as integrated one into the management system in the company that carries out investment activity. The functionality of software solution has been tested in the number of projects, realized by the Ukrainian investment companies.

References

1. Integration Definition for Function Modeling (IDEF0). Draft Federal Information Processing Standards Publication 183. USA: National Institute of Standards and Technology (1993)
2. Moskalenko, V.V., Zakharova, T.V., Dereza, E.O.: Technology of Decision Making for the Formation of Investment Policy of an Investment Company. Systems of Information Processing 9(90), 226–230, Kharkiv, Ukraine (2010) (in Ukrainian)
3. Zakharova, T.V., Moskalenko, V.V., Volovshchikov, V.Y.: Models and Techniques of Projects Portfolio Construction. Problems of Information Technology, 01 (007), 41-48, Kherson, Ukraine, in Russian (2010)
4. Zakharova, T., Moskalenko, V.: Modular Structure of Investment Decision Support System. In: Proceeding of The xth International Conference Modern Problems of Radio Engineering, Telecommunications and Computer Science, pp. 32–33. Publishing House of Lviv Polytechnic, Lviv (2010)
5. Power, D.J.: Decision Support Systems: Concepts and Resources for Managers. Greenwood, Quorum Books, Westport, CT (2002)
6. Demarest, M.: Technology and Policy in Decision Support Systems, DSSRecources.COM (2005)
7. Date, C.J.: An introduction to Database Systems. Addison-Wesley Longman, Inc. (2000)

Long-Term Forecasting Technology of Macroeconomic Systems Development: Regional Aspect

Marina Grinchenko, Olga Cherednichenko, and Igor Babych

National Technical University “Kharkiv Polytechnic Institute”,
21 Frunze str., 61002, Kharkiv, Ukraine
mgrinchenko@list.ru, marxx75@mail.ru, babich.igor.i@gmail.com

Abstract. Conducting of effective public policy is impossible without the scientifically substantiated evaluation of taken decisions. Long-term modeling and forecasting is the tool that allows evaluating the impact of the ongoing reforms on macroeconomic systems' development. This requires the processing of huge amount of diverse information. This article is devoted to modeling and long-term forecasting of regional macroeconomic systems' development. The simulation model based on the method of system dynamics is proposed. The generalized forecasting technology, which includes simulation model setup, validation and long-term forecasting, is developed. The numerical experiments and the forecast of Kharkiv region development until 2025 were carried out on the basis of the suggested information technology.

Keywords: forecasting, information technology, region, development.

1 Introduction

Modeling and forecasting of the complex objects' and systems' development is an integral part of decision-making process on different management levels. It allows to evaluate various aspects of object functioning and consequences of implementation of one or another management decision, and to elaborate a long-term development strategy.

Modeling and long-term forecasting of the territorial systems' development at the regional level is an important direction in this field of study. The complexity of the problem is stipulated by the variety of social, economical, and ecological processes that take place inside of the given systems, their interaction with the nearby territorial entities, dependence of the regional system upon the decisions made on the state level, and the environment.

The noted peculiarities of the regional macroeconomic system place limitations on the methods and means used to model the concerned object. It is obvious, that in order to conduct a comprehensive modeling of a regional system, a set of models having different nature and structure must be developed. This set of models should allow to forecast the main social, economic, and financial indices of the region's development, and to estimate the tax and expenditure, monetary, and administrative-territorial policies being pursued. However, the usage of the given models is associated with the

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 49–60, 2013.

collection, processing and storing substantial amounts of diverse socio-economic and financial information, received from different sources. It is impossible to solve this problem without implementing modern information technology.

Therefore, only the integration of the given models as a single information-computing environment will permit creating an effective technology for regional development forecasting, which can be used by the state or local authorities for elaborating the plan of development for the region, and estimating the impact of the ongoing reforms.

This paper is devoted to modeling and long-term forecasting of regional macroeconomic systems' development. The analysis of the existing forecasting methods is conducted in the second section. The third section is devoted to the set of mathematical forecasting models for the region's development, elaborated by the authors. The functional structure of applied information technology, based on the proposed models, is presented. The fourth section deals with the implementation of the suggested technology. The results of the regional system development forecast on the example of Kharkiv region until 2025 are provided. The fifth section contains conclusions on the conducted research and the directions of the further analysis.

2 Analysis of the Existing Approaches of Long-Term Forecasting of Macroeconomic Systems

Only two main "working" approaches can be distinguished from the variety of methods for modeling the structurally complicated macroeconomic systems. They are econometric [1,2] and simulation modeling [3,4].

Econometric models provide quantitative description of the interconnections between economic objects and processes, and they are developed for the forecasting of the economy's dynamics [2]. Modeling of the developed countries economies is often carried out with the help of econometric models. Wharton [5] and Brookings [6] models are especially popular.

Analysis of the econometric approach, based on the methods of modern applied statistics, shows that development and practical application of such models are based on the usage of rather simple, but highly effective methods and software tools, the elaboration degree of different applied aspects of which is quite high. The main disadvantages of these models are formal usage of regression analysis techniques which results in inadequacy of the models, and the limited forecasting interval which prevents from their usage for long-term forecasting.

Let's consider another group of methods including causal (simulation) models. The most known of them is the model of system dynamics [3,4]. J.W. Forrester developed the method of system dynamics and applied it to the examination of the processes of global economic development. The results, obtained by Forrester, were used in many models of world dynamics [7,8,9].

When analyzing the models and methods of world dynamics, some methodological disadvantages should be pointed out. For example, the territorial peculiarities are not taken into account. This leads to the complete disregard of social and economic differences while modeling the processes of world dynamics. As an advantage of the J.W. Forrester's model we can name the reflection of the main, natural connections between the elements of the world economic system, with the help of which these elements influence each other, determining the historical dynamics of the economic system. In [3] J.W. Forrester drew attention of the researchers to the significance of the objects behavior simulation with the help of computers.

One of the explanations of the vitality of the world dynamics models is their simplicity, and the fact that practically all the components of the model can be easily recorded and modified for the reason of substantial correlations, precisely formulated in economic and social terms.

The dynamics of the macroeconomic systems is characterized by the main groups of indices, which determine production sphere, consumption sphere and ecological sphere. The analysis shows that system dynamics model provide the largest amounts of information when used for describing development processes, which take place in macroeconomic systems. However, the conducted analysis of the existing forecasting methods shows that there are almost no theoretical solutions for complex long-term regional forecasting.

3 Long-Term Forecasting Technology

The main attention in the article is paid to the regional macroeconomic systems (RMES). The main stages of the long-term forecasting of RMES development process are shown in Fig.1. The main component of the suggested technology is the simulation model of RMES development processes (SMDP) [9]. The simulation model is based on the system dynamics, the levels of which are regarded as the first-type variables. In addition SMDP is supplemented by such variables as the society's intellectual capital (SIC), share of the intellectual capital in agriculture (SICA), and gross regional product (GRP).

Modern complex equipment and technologies cannot be used if there are no qualified workers. That is why for production simulation it is necessary to take into account the intellectual component. For this reason the SIC variable is used in the model. Similarly to the share of assets in agriculture in Forrester's model, the SICA variable is introduced. It reflects the level of usage of new techniques and technology in agriculture. The GRP is chosen as a variable because of the necessity to account the distribution of the GRP inside the system on the one hand, and on the other hand determines and formalizes the interaction with the environment.

Therefore, the SMDP RMES contains 8 first-type variables. The RMES development dynamics is determined based on the finite-difference equations

$$Y_1(\mathrm{t_k}) = Y_1(\mathrm{t_k}) + \Delta Y_1(\mathrm{t_k})\mathrm{h} \tag{1}$$

where alteration of each variable is determined by the rates of increase and decrease $\Delta Y_1(\mathrm{t_k})$, h is a discretization interval.

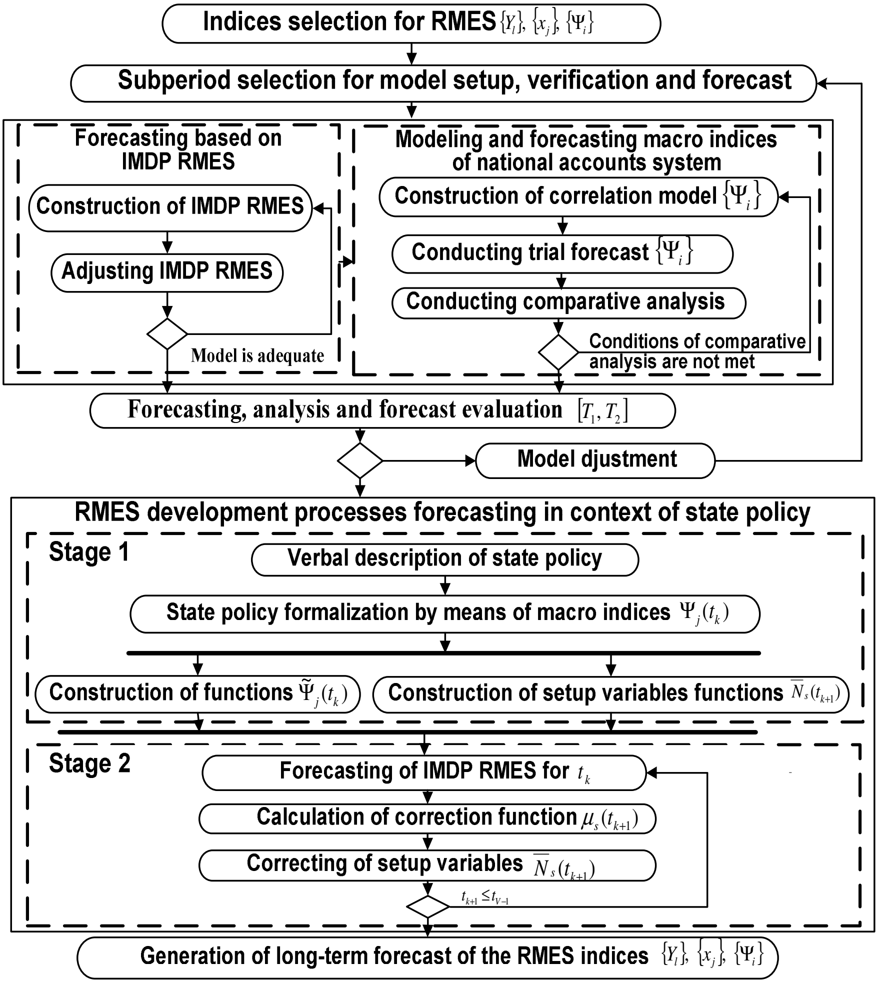


Fig. 1. Main stages of long-term forecasting of RMES development processes

The rates of increase and decrease are estimated for each first-type variable separately, and they represent the RMES processes that are characterized by the given variable. It should be mentioned, that GRP is modeled based on the production function.

Let's denote as $X(t_k) = \{x_j(t_k),\ j \in J\}$ the set of second-type variables that indicate the internal processes in RMES (Table 1). In order to describe the typical functioning of macroeconomic systems and setup the model to the actual RMES, the setup variables $N = \{N_1,..N_s..,N_v\}$, $s = \overline{1,v}$ are used (Table 2). In addition, table functions $q_i(x_j(t_k))$ are used, which describe causal relations in RMES.

Table 1. Second-type variables of SMDP RMES

№	Name of the second-type variable	Formula
1	Relative population density	$x_1(t_k) = Y_1(t_k) / N_1 \cdot N_2$
2	Relative value of assets	$x_2(t_k) = Y_2(t_k) / Y_1(t_k)$
3	Relative value of assets in agriculture	$x_3(t_k) = (x_2(t_k) \cdot Y_5(t_k)) / N_7$
4	Relative value of pollution	$x_4(t_k) = Y_4(t_k) / N_7$
5	Remainder of natural resources	$x_5(t_k) = Y_2(t_k) / N_{11}$
6	Relative level of population nutrition	$x_6(t_k) = (q_{18}(x_3(t_k)) \cdot q_{19}(x_1(t_k)) \cdot q_{20}(x_4(t_k)) \cdot q_{21}(x_{12}(t_k))) \cdot N_{14} / N_{13}$
7	Quality of life	$x_7(t_k) = q_{22}(x_1(t_k)) \cdot q_{23}(x_6(t_k)) \cdot q_{24}(x_4(t_k)) \cdot q_{25}(x_{10}(t_k)) \cdot N_{15}$
8	Material component of productive power level	$x_8(t_k) = (x_2(t_k) \cdot \dfrac{(1 - Y_5(t_k))}{(1 - N_7)} \cdot q_{17}(x_5(t_k))) / N_6$
9	Intellectual component of productive power level	$x_9(t_k) = (x_{11}(t_k) \cdot \dfrac{(1 - Y_7(t_k))}{(1 - N_{16})} \cdot q_{18}(x_5(t_k))) / N_{18}$
10	Level of production power development	$x_{10}(t_k) = (x_8(t_k) \cdot x_9(t_k))^{1/2}$
11	Relative value of society's intellectual capital	$x_{11}(t_k) = Y_6(t_k) / Y_1(t_k)$
12	Relative value of intellectual capital in agriculture	$x_{12}(t_k) = (x_{11}(t_k) \cdot Y_7(t_k)) / N_{16}$
13	Ratio of material and intellectual components of production power development level	$x_{13}(t_k) = x_8(t_k) / x_9(t_k)$
14	Share of capital spent on agriculture in terms of the existing nutrition level	$x_{14}(t_k) = q_{26}(x_{10}(t_k)) / q_{24}(x_6(t_k))$

Modelling algorithm of SMDP RMES functioning consists of the following steps.

Step 1. Setting of parameters initial values: first-type variables $Y_l(t_0)$, $l = \overline{1, M}$; second-type variables $x_j(t_0)$; setup variables N_s, $s = \overline{1, v}$.

Step 2. Calculation of amount of natural resources

$$Y_3(t_{k+1}) = Y_3(t_k) - Y_1(t_k) \cdot N_5 \cdot q_{12}(x_{13}(t_k)).$$

Step 3. Calculation of assets

$$Y_2(t_{k+1}) = Y_2(t_k) + (Y_1(t_k) \cdot N_8 \cdot q_{11}(x_{10}(t_k)) - Y_2(t_k) \cdot N_{10}).$$

Step 4. Calculation of environmental pollution level

$$Y_4(t_{k+1}) = Y_4(t_k) + (Y_1(t_k) \cdot N_{12} \cdot q_9(x_2(t_k)) - Y_4(t_k) \cdot (q_{10}(x_{13}(t_k)))^{-1}).$$

Step 5. Calculation of population size

$$Y_1(t_{k+1}) = Y_1(t_k) + (Y_1(t_k) \cdot N_3 \cdot q_1(x_1(t_k)) \cdot q_3(x_6(t_k)) \cdot q_5(x_4(t_k)) \cdot q_7(x_{10}(t_k)) - \\ - Y_1(t_k) \cdot N_4 \cdot q_2(x_1(t_k)) \cdot q_4(x_6(t_k)) \cdot q_6(x_4(t_k)) \cdot q_8(x_{10}(t_k))).$$

Step 6. Calculation of share of assets in agriculture

$$Y_5(t_{k+1}) = Y_5(t_k) + (q_{13}(x_6(t_k)) \cdot q_{14}(x_{14}(t_k)) \cdot (N_7)^{-1} - Y_5(t_k) \cdot (N_7)^{-1}).$$

Step 7. Calculation of SIC

$$Y_6(t_{k+1}) = Y_6(t_k) + (Y_1(t_k) \cdot q_{15}(x_{10}(t_k)) \cdot N_{19} - Y_6(t_k) \cdot N_{20}).$$

Step 8. Calculation of SICA

$$Y_7(t_{k+1}) = Y_7(t_k) + (q_{16}(x_6(t_k)) \cdot q_{14}(x_{14}(t_k)) \cdot (N_{16})^{-1} - Y_7(t_k) \cdot (N_{16})).$$

Step 9. Calculation of GRP

$$Y_8(t_k) = \alpha_0 K_w Y_1^{\alpha_1}(t_k) \cdot Y_2^{\alpha_2}(t_k) \cdot Y_6^{\alpha_3}(t_k).$$

Step 10. Calculation of the second-type variables values for the t_{k+1} year of simulation.

Step 11. System time is increased, k=k+1. If k ≤ V-1, then move to Step 2, otherwise simulation process is stopped.

Table 2. Setup variables

Symbol	Description	Symbol	Description
N_1	area, km2	N_{12}	average rate of pollution generation, c.u./year
N_2	average population density, ppl/km2	N_{13}	average nutrition rate, c.u.
N_3	average birth rate, ppl/year	N_{14}	nutrition coefficient, c.u.
N_4	average death rate, ppl/year	N_{15}	standard quality of life, c.u.
N_5	average natural resources production, c.u./year	N_{16}	average intellectual share in agriculture, %
N_6	average efficiency of relative value of assets, c.u./ppl	N_{17}	time of turnover of intellectual assets in agriculture, year
N_7	average share of assets in agriculture, %	N_{18}	average efficiency of relative intellectual value, c.u./ppl
N_8	average rate of assets generation, c.u./year		
N_9	time of turnover of agricultural assets, year	N_{19}	average rate of intellect generation, c.u./year
N_{10}	average rate of assets depreciation, c.u./year	N_{20}	average rate of intellect consumption, c.u./year
N_{11}	average environmental pollution, c.u./year	N_{21}	initial natural resources supply of macroeconomic system, c.u.

The technology of RMES development process forecasting is considered on the three time intervals: setup, validation and forecasting. Verification of the model's adequacy, which is conducted during SMDP setup (Fig.1), is an important aspect in simulation modeling. During the setup the initial values of all model's parameters are determined. It should be noted that all the values of table functions are estimated either through scaling of the corresponding values of system dynamics model, or with the help of expert methods. Setup variables Ns are the main tool for adjusting the model to the real macro environment. The sufficient level of SMDP adequacy, based on the Theil and Gini coefficients for the first-type variables and the entire model, is the setup success criterion.

Since the system dynamics method does not provide the full description of the RMES, the forecasting technique includes the stage of forecasting the indices of national accounts system (NAS). It provides additional indices for RMES evaluation. In order to complete this stage, correlation analysis of the set of macro indices $\Psi_i(t_k),\ i=\overline{1,\lambda}$ is conducted and their interrelation is determined. Based on the forecast values $\Psi_i(t_k),\ i=\overline{1,\lambda}$ the GRP is calculated using the macro agreement formula. After that the consistency of the forecast is evaluated based on the Gini coefficient. Such procedure provides the balanced forecast for all indices, based on the SMDP and the NAS. The set up forecasting model is used for formation of the trial forecast on the test interval. The Gini coefficient is suggested as a success criterion.

The forecasting of impact of the state policy will be conducted through SMDP RMES. In order to describe the effect of the state policy on RMES, the simulation model should be complemented by the following items: 1) the model, which is created based on the state policy formalization; 2) the setup variables correction model.

At the first stage the initial conditions of the government control are formed. Then the SMDP RMES setup variables $N_s(t_k)$, sensitive to the chosen policy, are determined as the time functions that depend on the change of $\tilde{\Psi}_P$ macro indices of the state policy. At the second stage the forecasting of the main SMDP RMES variables is conducted. Based on the obtained results, the calculation of the corresponding parameters $N_s(t_k)$ is carried out. At each step the values of parameters $N_s(t_k)$ are transmitted to the SMDP RMES in order to forecast the following period.

After finishing the forecasting for different control actions, the reports are formed.

It is suggested to use the developed set of models in analysis information system of RMES development processes forecasting (Fig.2).

4 Research Results

Let's consider the use of the developed applied information technology for forecasting the development processes of Kharkov region. The period from 2000 to 2006 was chosen in order to setup the model. During this period, the initial values of 8 model variables and 21 setup variables were determined, and 25 table functions of causal relationship were constructed with the help of statistical data concerning social and economic development.

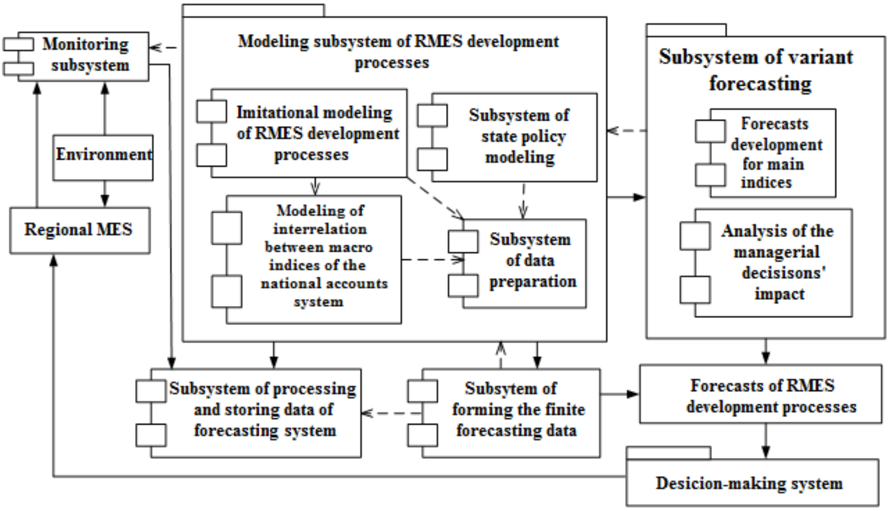


Fig. 2. General system structure of RMES development processes forecasting

For constructing the table functions, the corresponding factors of the Forrester's world dynamics model and expert judgments of region's development dynamics were used.

After setting all necessary initial data, trial forecasting was conducted. Comparison between the forecast results and actual indices of social and economic development of Kharkov region during the years 2000-2006 was carried out by means of calculating Theil and Gini coefficients. Table 3 shows that under the initial values of setup variables the forecast data substantially deviate from the actual data.

Table 3. Evaluation results of forecast adequacy during the period of 2000-2006

Level	Definition	Initial values of N_s		Finite values of N_s	
		Theil coefficient	Gini coefficient	Theil coefficient	Gini coefficient
$Y_1(t_k)$	Number of population	0,007	0,001	0,000	0,001
$Y_2(t_k)$	Amount of existing assets	0,019	0,002	0,007	0,001
$Y_3(t_k)$	Amount of natural resources	0,012	0,001	0,005	0,001
$Y_4(t_k)$	Environmental pollution	0,058	0,005	0,007	0,001
$Y_5(t_k)$	Share of assets in agriculture	0,398	0,052	0,015	0,002
$Y_6(t_k)$	Society's intellectual capital	0,010	0,001	0,002	0,001
$Y_7(t_k)$	Share of intellectual capital in agriculture	0,410	0,054	0,011	0,001
$Y_8(t_k)$	Gross regional product	0,136	0,017	0,058	0,006
Entire model		0,159	0,106	0,019	0,011

Deviation for the share of assets in agriculture equals to 39,8%, intellectual share in agriculture – 41%, gross regional product – 13,6%, and from the entire model – 15,9%.

In order to improve the forecast accuracy and adequacy, a set of setup variables was determined. This set was obtained by means of altering every setup variable by ±5%, ±10%, and ±50%, and calculating Theil and Gini coefficients after each experiment. After 200 model runs the set of setup variables was complemented by such indices: nutrition coefficient N14, average level of nutrition N13, average shares of intellectual capital N16 and of assets N7 in agriculture.

The given parameters have substantial influence on the level of share of assets in agriculture and the level of intellectual capital in agriculture. The levels of these indices had considerable deviations during initial model adequacy verification. It was possible to achieve the admissible estimate of adequacy of trial forecast results during the setup period by means of altering the values of these 4 parameters. Consequently, the deviation of the entire model was 1,9% for Theil coefficient and 1,1% for Gini coefficient. Therefore the model setup was finished.

The adjusted simulation model was further used for forecasting the development processes of Kharkov region for the period until 2025. The year 2007 was chosen as the starting point. In order to assess the forecast accuracy for the subperiod 2007-2010, the statistical information was collected and the indices values were determined. The accuracy evaluation of the trial forecast was conducted similarly to the setup of the model that is by means of calculating Theil and Gini coefficients.

The results of the trial forecasting are shown in Fig.3. The black column in the figure is the actual value of the index, and the grey column is the forecasted one. All values of Theil and Gini coefficients, calculated for separate values as well as for the entire model during the subperiod 2007-2010, lie between 0,8% and 1,2%. These values show the considerably high level of forecast accuracy.

The forecasting results of Kharkov region development for 2011-2025 are shown in Fig.4. They indicate that under the provided current state policy, the number of population will decrease to 2.57 million people; the amount of existing resources will decrease 1,5 times; the existing assets will increase 1,4 times; average growth rate will be 2,5% per year; the share of assets in agriculture will decrease by 20%; the level of environmental pollution in the region will slightly decrease; the GRP will grow at the rate of 1,5-2% annually.

To analyze the forecast it is necessary to take into account the fact that the GRP growth is stipulated mainly by the increase of capital assets. The growth of capital assets can be explained by the fact that the rest of capital assets is not utilized, and the new expensive ones are imported. Hence, the perspective of region's development is not very auspicious and it is necessary to immediately increase the efficiency of development processes management in the region.

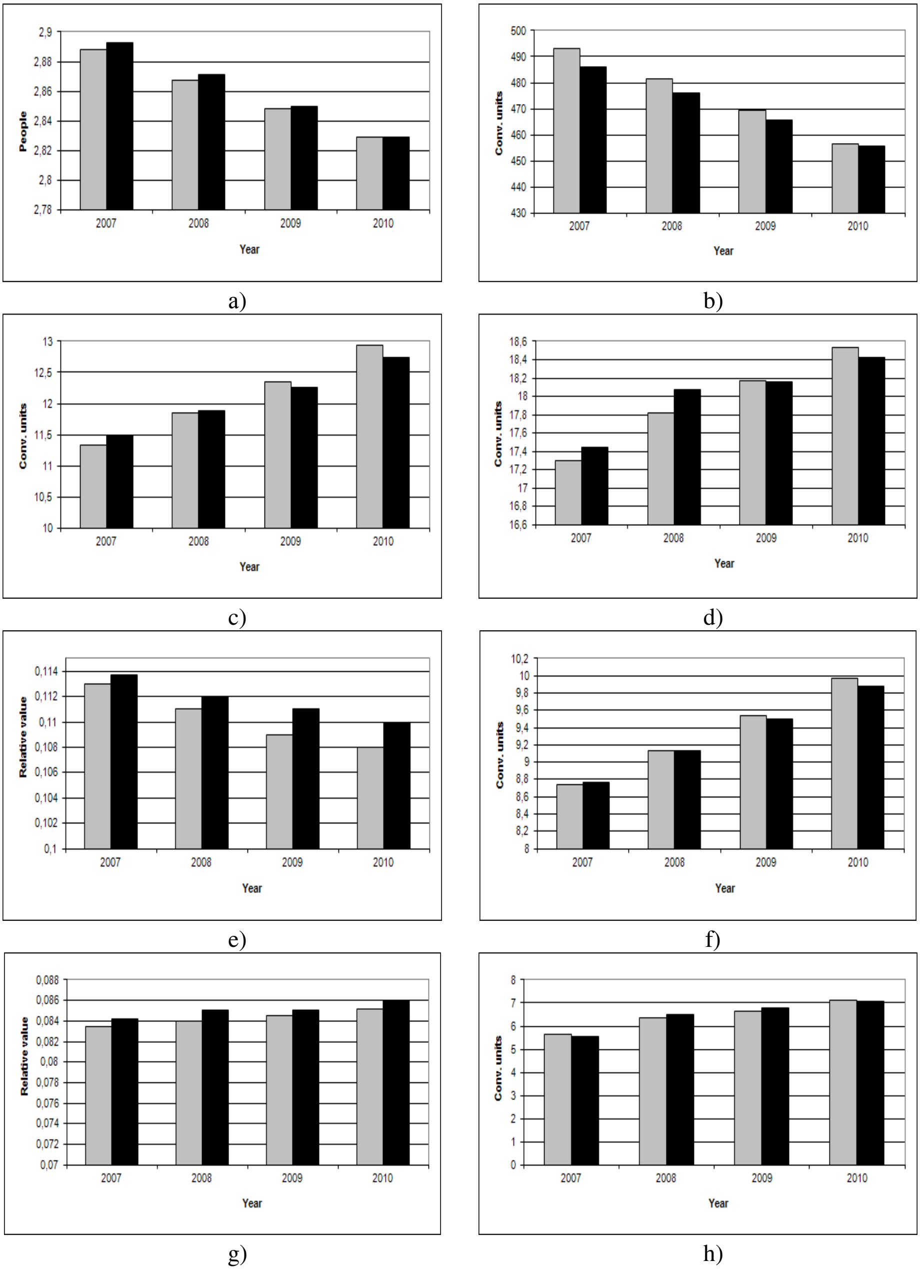


Fig. 3. Forecast results during the subperiod 2007-2010: a) Y_1, b) Y_2, c) Y_3, d) Y_4, e) Y_5, f) Y_6, g) Y_7, h) Y_8

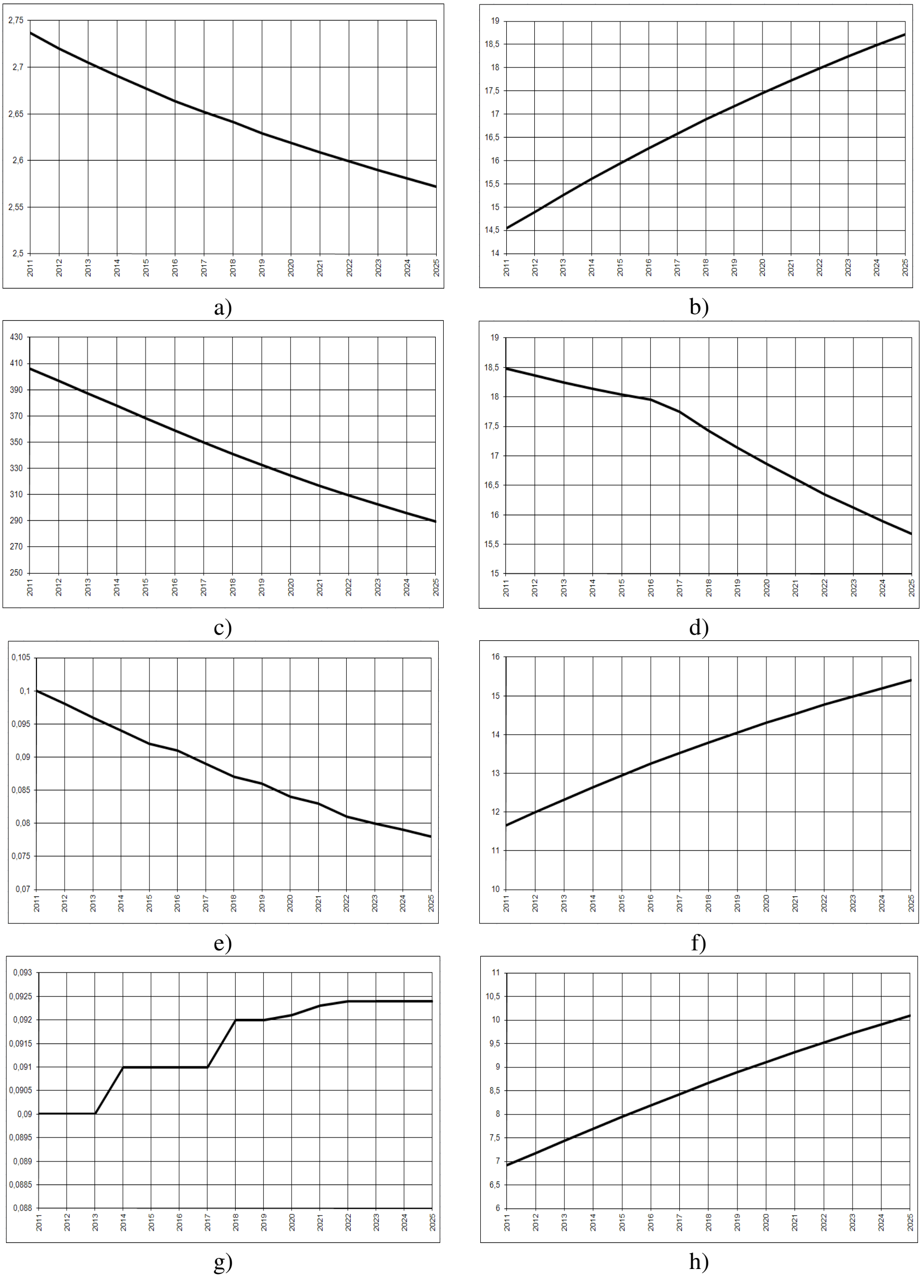


Fig. 4. Forecast of Kharkov region development until 2025: a) Y_1, b) Y_2, c) Y_3, d) Y_4, e) Y_5, f) Y_6, g) Y_7, h) Y_8

5 Conclusions and Future Work

Analysis of complex social and economic systems requires taking into account the peculiarities of systems of such class. These complex systems contain interconnected relationship between people and processes which cannot be described in quantitative way. As a rule, the consequences of present actions are delayed in time. Also there are many factors that may take different values in future.

Scenario approach allows studying different variants of system's development in future and elaborating flexible strategy of efficient state policy deployment in some region. The suggested in the given work information technology is a useful and reasonable tool for implementation of scenario approach. It allows to include alternative variants of scenarios in the model of RMES development processes and to estimate their impact on the long-term perspective. Therefore the further researches will be aimed at the development of the forecasting technology in the context of state policy and elaboration of scenario forecasting technology.

References

1. Klein, L.R., Young, R.M.: An Introduction to Econometric Forecasting and Forecasting Models. Lexington Books (1980)
2. Allen, P.G., Fildes, R.A.: Econometric forecasting strategies and techniques. In: Armstrong, J.S. (ed.) Principles of Forecasting: A Handbook for Researchers and Practitioners. Kluwer Academic Press, USA (2001)
3. Forrester, J.W.: World Dynamics. Wright-Allen Press, Cambridge (1971)
4. Meadows, D., Randers, J., Meadows, D.: Limits to Growth: The 30-Year Update., White River Junction, VT, Chelsea Green (2004)
5. Evans, M.K., Klein, L.R.: The Wharton Econometric Forecasting Model, 2nd edn., Philadelphia (1968)
6. Duesenberry, J.S., Fromm, G., Klein, L.R., Kuh, E.: The Brookings Model: Some Further Results, Chicago (1969)
7. Forrester, J.W.: Urban Dynamics. Pegasus Communications, Waltham, MA (1969)
8. Turner, G.: A comparison of the Limits to Growth with 30 years of reality. Global Environmental Change 18(3), 397–411 (2008)
9. Lisitskiy, V.L., Grinchenko, M.A.: Working out of simulation forecasting model of developments of macroeconomic systems. Eastern European Journal of Enterprise Technologies 3/5(39), 4–8 (2009)

Using Lightweight Activity Diagrams for Modeling and Generation of Web Information Systems

Dirk Reiß[1] and Bernhard Rumpe[2]

[1] Institute for Building Services and Energy Design
Technical University of Braunschweig, Germany
http://www.igs.bau.tu-bs.de
[2] Software Engineering
RWTH Aachen University, Germany
http://www.se-rwth.de

Abstract. The development process of web information systems nowadays improved a lot regarding effectiveness and tool support, but still contains many redundant steps for similar tasks. In order to overcome this, we use a model-driven approach to specify a web information system in an agile way and generate a full-fledged and runnable application from a set of models. The covered aspects of the system comprise data structure, page structure including view on data, page- and workflow within the system as well as overall application structure and user rights management. Appropriate tooling allows transforming these models to complete systems and thus gives us opportunity for a lightweight development process based on models. In this paper, we describe how we approach the page- and workflow aspect by using activity diagrams as part of the agile modeling approach MontiWIS. We give an overview of the defined syntax, describe the supported forms of action contents and finally explain how the behavior is realized in the generated application.

Keywords: Web Information Systems, Workflow, Activity Diagrams, Domain Specific Language, Modeling.

1 Introduction

Even though the development of web information systems is supported by a variety of web frameworks in almost all modern programming languages (see [1] for an overview of those), it still requires a lot of repetitive, tedious and error prone work for rather basic tasks like the definition of data structure and the creation of CRUD (create, read, update and delete) functionality with according pages therefor or basic but system-wide consistent user and rights management. To alleviate the effort to implement a web information system, we developed the model-driven approach MontiWIS to abstract from the details of the repetitive implementation tasks while maintaining flexibility where necessary. Following an agile development approach, we allow rapid prototyping by providing extensive

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 61–72, 2013.

default behavior for a minimal set of models and stepwise refinement of the applications' functionality through the addition of further models and aspects to the system.

As the data model is the basic concept in our approach, a developer can start his application model by only specifying this part of the system. The generation process results in a full-fledged runnable system with default CRUD functionality for the given data model. From this point, further aspects like user- and rights management or special views on the data model which in turn can be incorporated in complex business processes can be specified. As processes and business logic are a crucial part of a web information system, we focus on this aspect in the following and describe the different development stages from the specification using a profile of UML activity diagrams [2] to the technical realization in the generated application.

The paper is structured as follows. In Section 2, we shortly describe the MontiWIS approach in a whole and explain the interdependency of the different languages and aspects of the system. Section 3 introduces a graphical example, describes its equivalent in the textual syntax of activity diagrams and explains the supported action contents. In Section 4, we give an overview of the technical realization of processes in the generated system. In Section 5, similar approaches are discussed and Section 6 concludes the paper and gives an outlook on planned extensions of our approach.

2 Overview of MontiWIS Development Method

MontiWIS is the successor of MontiWeb [3] and extends it especially in the area of user- and rights management, application construction and page description expressivity. Both approaches use MontiCore [4,5] and its infrastructure to define the abstract and concrete syntax of their textual languages and to process the models.

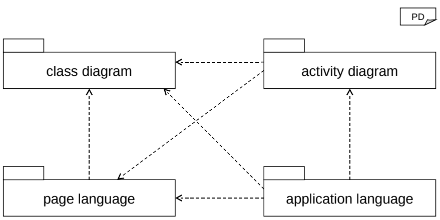


Fig. 1. Languages and their relation in MontiWIS

The package diagram in Figure 1 shows the different models that are used to specify structure and behavior of the application. Each package in the figure depicts one modeling language that covers a certain aspect of the system. An arrow from one language to another indicates that parts of the targeted language are referenced by the source language. The UML/P [6,7] class diagrams, which are used to describe the data structure of the web information system, are based on the grammar and infrastructure of the work by Martin Schindler [8]. As an addition to his implementation, we offer domain specific types like for instance String, Email, Date and Text that result in the generation of type-specific input elements with consistency checks and a reasonable default behavior. The domain specific page language facilitates the description of a user interface page with elements like, among others, headings, simple text parts and complex tables. Pages may have input parameters (whose types are usually classes from the class diagram) that can be displayed in a whole or attribute-by-attribute in either an editable (i.e. using input forms) or non-editable way (by just outputting their values). The activity diagrams are used to describe the control- and data flow between pages as well as complex multi-user workflows. An action, being the executable part of an activity, may contain references to a page that will be displayed when an action is executed. It may also reference classes and operate on them as part of the included business logic. This way, we support both, actions that require user interaction through page display as well as completely automatic ones that are executed on the server in a standalone manner. However, in most cases, code is executed before and after the presentation of a page. Section 3 describes the language in detail. Finally, the domain specific application language serves as central model to define flexible roles and associated rights (e.g., guest, student or administrator) in the application. It allows the definition of the systems' navigation menu and, considering different roles, different views on the application as a whole. Elements of all three languages can be referenced and included as menu entries or can have rights restrictions applied to them. Classes are referenced implicitly by including default pages for creating new or listing all objects of a certain type, static pages can be included to display predefined content and activities are referenced as workflows that can be invoked.

By using different languages for different aspects of the system, we can modify one aspect (such as user roles or page structure) without the need to adapt related others (e.g., activities that may be invoked by a certain role or ones that use the page). Although all these languages are defined independently, the MontiCore infrastructure easily allows us to define and check connections between these languages. This way we can for instance check the existence of classes referenced as data types in either pages or activities, check whether attributes are used correctly or ensure that pages and activities are linked correctly in menu and rights declarations.

As our lightweight approach is designed with agile methodologies in mind, reasonable default behavior is generated in the final application where specific details are omitted. This approach allows for early results while maintaining flexibility where needed.

Currently, the MontiWIS generator creates a full-fledged Java EE application that uses JPA for persistence mechanisms and JSP pages (with included JQuery on the client side) to create the modular and AJAX-enabled user interface of the application.

3 Page- and Workflow Using Activity Diagrams

In order to explain the principles and possibilities of activity diagrams in MontiWIS, we first introduce the example given in Figure 2. Afterwards, we explain how the described example can be specified by using the language and constructs of our approach.

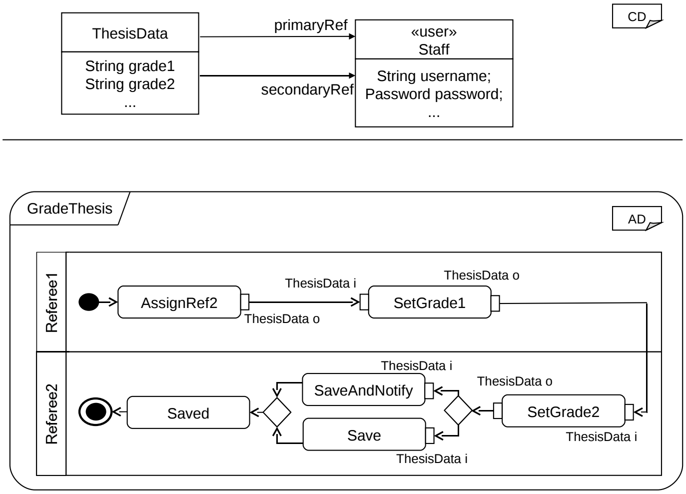


Fig. 2. Example: class and activity diagram to grade a thesis

The diagrams show an excerpt of a system that supports a lecturer in his curriculum tasks. The relevant part of the underlying data structure is given in the upper half. It consists of only two classes, ThesisData and Staff, each with just a subset of the attributes included here. Besides two attributes that hold grades (grade1 and grade2), ThesisData includes two associations to Staff that represent the thesis' referees (primaryRef and secondaryRef). Staff is stereotyped with user, which indicates that this domain model class also holds information to log in to the system and that objects of this class can serve as actors in a workflow.

The activity diagram in the lower part of the figure shows the workflow that guides two different participants through the process of assigning grade values to a thesis. As Referee1 is the partition that contains the first action in the workflow, the person starting the activity is automatically assigned to it. In the first action of the activity (AssignRef2), one Staff is selected from a list of all possible ones and then assigned to partition Referee2. In order to complete the ThesisData object passed on along the workflow (indicated by pin o), both, the Staff actually executing the workflow and the one selected within the action, will need to be set as well. Afterwards, the ThesisData object is handed over to the next action SetGrade1, also conducted in the same partition and thus by the same user. In this action, the user determines and assigns a value to attribute grade1 in the given object. After this step, no further tasks need to be executed by the first referee and the ThesisData object is passed on to action SetGrade2 that is associated with the other partition. Therefore, this workflow will appear in the list of active workflows of the user previously associated with this partition. Just as the Referee1, Referee2 also needs to assign his grade to the thesis data, which he is asked to do in action SetGrade2 when he resumes the activity. At this point, the workflow is split up in two alternative directions, either of which he can take. As there are no guards that could be evaluated automatically, the user will be prompted for the desired direction. In our approach, each possibility is represented by a button that is included in the page at execution time. The subsequent action SaveAndNotify persists the object in the database and sends a notification to all participants while Save only persists the object. After either of these actions, the action Saved notifies the user of the successful action by displaying an appropriate page. A final submission of this page finishes the workflow and cleans up runtime data.

As already mentioned in Section 2, we use a textual syntax to describe the system. This also applies to the models described above. For the sake of brevity and focus of this paper, we omit the textual representation of the given class diagram and only describe (parts of) the activity diagram language in Figure 3.

The diagram itself is identified by the keyword `activity` and a unique name (1). A partition of an activity is defined by the keyword `role`, its name and a set of actions that are associated with them in curly braces (3). An action is denoted by the keyword `action` and a name that has to be unique within the activity. The actions' content is included in braces (6-20). In case an action contains in- or output parameters (or both), we declare them at the beginning of the action (7, 25). The keywords `in` and `out` denote the type of the parameters that are listed after a colon and separated by a comma. A parameter declaration consists of its type (which is usually defined in the class diagram or one of a few allowed Java data types such as `Set` (8)) and a name that is scoped within the action and thus can be used throughout its contents. The same applies to variables (denoted by the keyword `var` (8)), which can be used within the action but are not accessible from outside of it.

The business logic within an action is executed in the order it is defined. Here, we first invoke several predefined commands. A command consists of the keyword `cmd` and a colon, followed by the syntactically predefined command itself. These commands are specific to the domain of web information systems and provide functionality such as database access, or mechanisms for workflow control. The command in line 10 for instance retrieves a set of all `Staff` objects from the database and assigns them to the previously declared variable `allStaff`. The command in line 11 allocates the variable `actualUser` with the actually logged in user object (which is of type `Staff`), which then can be accessed in the remainder of the action.

Activity diagram

```
1  activity GradeThesis {
2
3    role Referee1 { AssignRef2, SetGrade1, ... }
4    // ...
5
6    action AssignRef2 {
7      out : ThesisData o;
8      var : Set<Staff> allStaff, Staff actualUser, Staff selectedUser;
9
10     cmd : allStaff = Staff.loadAll();
11     cmd : actualUser = getActualUser();
12     view : SelectSecondaryRef(allStaff);
13     java : {
14       selectedUser = allStaff.iterator().next();
15       o = new ThesisData();
16       o.setPrimaryRef(actualUser);
17       o.setSecondaryRef(selectedUser);
18     }
19     cmd : assignRole(Referee2, selectedUser)
20   }
21
22   action SetGrade1 { ... }
23
24   action SetGrade2 {
25     in : ThesisData i;
26     out : ThesisData o;
27
28     view : SetGrade2Page(i);
29     java : {
30       o = i;
31     }
32   }
33
34   // other actions omitted here
35
36   initial -> AssignRef2;
37   AssignRef2.o -> SetGrade1.i;
38   SetGrade2.o -> SaveAndNotify.i | Save.i;
39   // other edges omitted here
40 }
```

Fig. 3. Activity diagram in textual syntax

The keyword `view` (12) indicates the presentation of a page as part of the actions' logic. The page `SelectSecondaryRef` is defined in a separate model and accepts a set of `Staff` objects as parameter. Pages generally accept one or more objects as input and return them with updated values after submission of the page. In this case, the page contains a listing view of the given set of objects and allows the selection of a single object. Therefore, the variable `allStaff` contains a single `Staff` element after line 12.

In order to allow as much flexibility in declaring business logic as possible, we allow plain Java code to be included in an action. A block of code begins with the keyword `java`, followed by a colon and surrounded by braces (13-18). Inside them, arbitrary code may be included. Here, we first retrieve the selected `Staff` object from the submitted set of objects (14), followed by the instantiation of a new `ThesisData` object (15) and the assignment of both, the actual and the manually selected `Staff` to the domain object (16-17). Finally, the selected `Staff` object is assigned to the role named `Referee2` (19).

The action `SetGrade2` (24-32) contains in- and output parameters and will display the object assigned to the input parameter through page `SetGrade2Page` (26). This page is defined in a way that only one of the two grade attributes is editable and thus only the correct attribute can be updated. Finally, the submitted `ThesisData` is assigned to the output parameter (30). As described in the example before, the action `SetGrade1` facilitates a similar behavior, just different parts of the object are editable (by a different user). The remaining actions are omitted here due to lack of space.

Lines 36-38 show some of the possible ways to define activity edges. An edge is defined by a source node (and optional output parameter, separated by a dot) on the left hand side of an arrow (`->`) and the target node (and possible input parameter) on its right hand side, all identified by their node and parameter names. Line 36 defines the edge from the activities' initial node (denoted by the keyword `initial`) to action `AssignRef2`. Line 37 defines an edge from the output parameter of action `AssignRef2` to the input parameter of action `SetGrade1`. Line 38 shows an abbreviated syntax for a decision node (indicated by the pipe symbol (`|`)). Here, the output parameter of `SetGrade2` is connected with the input parameter of either `SaveAndNotify` or `Save`.

This activity diagram demonstrates a subset of the features that we offer to model business logic in our web information system. It provides both, flexibility by including Java code directly but also domain specific functionality like the presentation of pages, access to database and server runtime information as well as activity execution information. When designing the language, we focused on compact syntax and preferred to include pure Java syntax to defining a separate action language that would cover the same functional range. As we actually focus on Java as a target platform for generated code, we can easily include existing functionality from, e.g., external APIs or manually implemented complex logic almost instantly by invoking them from within an action. Nevertheless, other languages (like, e.g., C#, PHP or an abstract action language as used in executable UML [9]) could be included. Furthermore, the used Java code could as well be used to generate code in the above-mentioned languages, which would require a more complex code generator and appropriate mappings from Java constructs to corresponding ones in the desired target language.

4 Technical Realization

After describing how workflows and included business logic are specified in MontiWIS, we now focus on the technical realization of these aspects in the generated system. The execution follows the semantics we defined in [10] and thus the behavior given by the UML itself. As we wanted control over the way we execute both, simple page- and complex workflow, we preferred to implement the execution components ourselves instead of using freely available workflow engines such as Activiti [11] or JBPM [12] whose integration needed rather complex changes to allow the behavior we envisioned. A future integration of such existing engines is possible but for the time being not planned. Figure 4 shows a simplified overview of the architecture used to process and execute the workflows.

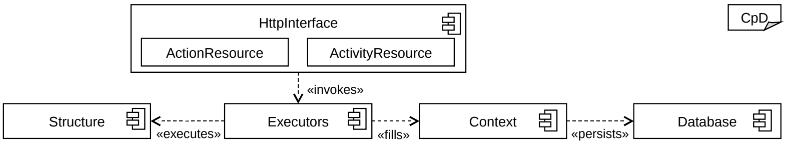


Fig. 4. Execution architecture of activities

As a basic principle, MontiWIS follows a REST [13] approach when interacting with the application. This implies that basic constructs of the application are treated as resources that are addressed by URLs and manipulated by HTTP requests. Besides the business domain objects (specified by the class diagram), we also treat activities and actions as resources. The two classes ActionResource and ActivityResource (part of the component HttpInterface) are accepting requests to both, actions and activities. The former is used to interact with a workflow once it has been started, the latter provides access to activities that need to be started, stopped or continued. The component Structure consists of the classes that are generated from the input models. They contain model-specific information such as activity structure or business logic in the case of an action. The component Context holds information about execution state such as actual values of node parameters or some sort of token representation to reflect the overall activity state. These classes are persisted in a database and thus can be retrieved at a later point to continue system execution or to share process state among different users of the application. The Executor component merges the runtime information from the contexts with the structure of the actual activities. Its classes include logic to control activity execution and update state, always specific to the type of the node. Each of the components contains classes with node-specific functionality for different types of nodes in an activity, such as for instance actions or decision nodes.

In order to explain typical steps of activity execution, we illustrate the different requests and responses occurring by means of the actions of Referee2 in Figure 2. The sequence diagram in Figure 5 shows HTTP requests and how they are processed in our application.

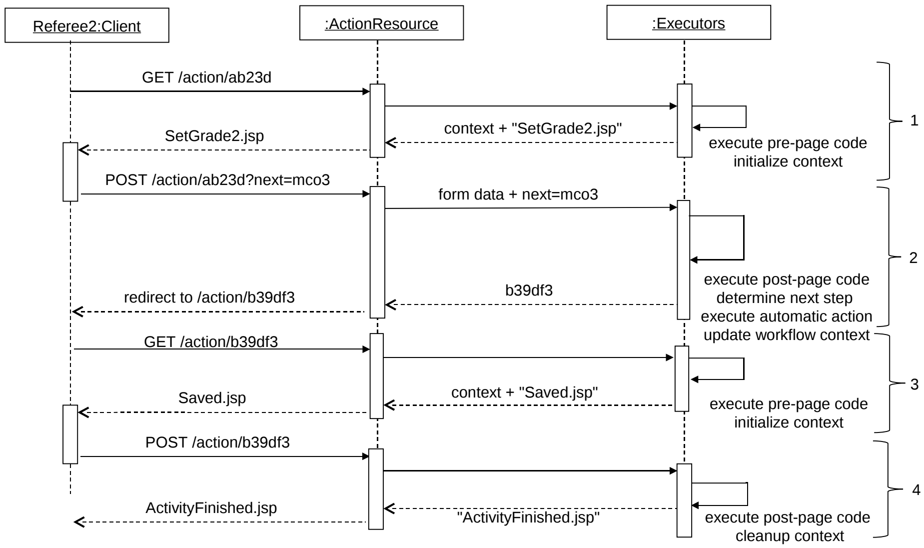


Fig. 5. Sequence diagram of HTTP interaction with workflows

The participating actors in the sequence diagram are the Client that interacts with the system through a web browser, the ActionRessource (which is invoked through an URL that addresses actions through the part `/action` as depicted in the figure) that receives the HTTP requests and returns the response to the client. The last actor (labeled Executors for the sake of simplicity) represents the different executor classes involved in the actual processing.

As we mentioned earlier, the execution of an action is usually split into a block of code executed before the display of the page, the page presentation itself and the code executed after the submission of possible form data. Step 1 in Figure 5 is initiated by a GET request to an action identified by the appended id (`ab23d` in this case). The resource class calls the executors to process the code defined before the page presentation and to initialize the page context, which contains all necessary information such as the ThesisData to display. Both, the context and the SetGrade2 JSP page itself are returned to the resource class and, after the execution of the JSP code, sent back to the client. As described in Section 3, decision nodes that do not contain any guards are interpreted as a free choice for the user of the application. Thus, the generated and presented page will contain elements that let the user choose which path to take after page submission. As shown in step 2, additionally to the form data, information about the users' decision is also sent to the resource class. The form data as well as the chosen path are passed to the executors that process potential post-page code, determine the next page based on the user decision and update the activities runtime context. As the next action is a completely automatic one that does not contain any user interaction whatsoever, the adequate logic will be executed, which is also part of the executors in step 2. Afterwards, the activity state is updated again and the identifier of the next action that also contains user interaction is returned to the ActionResource class. The identifier is used to address the next action and a HTTP redirect is send to the user that points to this action. In step 3, a GET request addressing the redirection target is automatically issued by the users' browser, the pre-page code is executed and a page indicating a successful save operation is displayed. Although the next step in the activity is the final node, we still demand another page submission in order to trigger possible post-page code to be executed. This call is depicted in step 4. Finally, the execution context is cleaned up and removed from the database.

5 Related Work

In the context of model driven development of web information systems, several approaches exist that also offer support for page- and workflow, most of them exploiting graphical models to describe the system. An evaluation of different web modeling approaches can be found in [14].

WebML [15] offers workflow concepts by incorporating process execution elements in their global model and, in contrast to our approach, do not offer a separate kind of model. Business logic is basically specified through a set of parametrized operation elements which does not offer the flexibility of plain Java code as MontiWIS allows.

WebWorkFlow [16] employs a textual domain specific language to describe processes with different aspects like, e.g., user access control or page references included in the process model. Here, model transformation approaches are used to transform higher level constructs to the more basic ones of WebDSL [17]. Unlike this approach, we do not use the basic concepts of some underlying domain specific language but rather our executor classes to enforce the process execution semantics.

The UWE approach [18] and especially its application UWE4JSF [19] are based on pure UML profiles. This approach uses annotated and stereotyped class diagrams to describe basic navigation and, although activity diagrams are also allowed to specify process logic, does not offer rich support for distributed processes. Furthermore, it focuses on atomic method calls and does not provide support for a close integration of user interaction and business logic.

The effort to integrate process execution into OOWS [20] uses BPMN as notation and model transformations to generate an extended OOWS navigation model (which itself is transformed to web pages) as well as WS-BPEL code to execute the process logic externally. Thus, in contrast to our approach, the process is not an integral part of the system but runs outside of the normal application logic.

OOHDM [21] offers means to tightly integrate processes and business logic in the navigational and conceptual models and does not allow separate models as MontiWIS does. Furthermore, complex business logic is rather hard to integrate, as interfaces to external program resources are not easily accessible.

Most of these approaches require a complete set of models covering all aspects of the system to generate the application and, unlike MontiWIS, do not provide extensive default behavior where parts of the application models are missing.

6 Conclusion and Future Work

In this paper, we described the business logic and workflow aspect of our modeling approach to alleviate the effort to develop web information systems. We shortly gave an overview of the basic concepts of our approach and how they are interrelated. We then introduced an illustrating example and gave a subset of the textual notation that we use in our system, highlighted how we interpret the execution of actions and activities and what possibilities we offer to specify action contents. Afterwards, we gave a simplified overview of our server side architecture regarding activity execution and described how typical workflow execution is carried out between a client and our server components.

Although the actual state of our implementation gives us great flexibility to express almost arbitrary business logic, we still see some areas of improvement. One actual focus of our work is the usability of our approach. This encompasses the development of workflow libraries that exploit hierarchical decomposition of activities to extract components that can be reused in other contexts. For the same reason, we plan to implement more commands to offer convenience functionality such as email transport or more fine-grained database search capabilities. Such functionality can actually be incorporated through plain Java calls but as our experience shows, could use closer integration with the generated system and runtime environment.

References

1. Vosloo, I., Kourie, D.G.: Server-Centric Web Frameworks: An Overview. ACM Computing Surveys 40(2), 1–33 (2008)
2. Object Management Group: OMG Unified Modeling Language (OMG UML), Superstructure Version 2.4.1 (11-08-06) (August 2011), `http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/`
3. Dukaczewski, M., Reiss, D., Rumpe, B., Stein, M.: MontiWeb - Modular Development of Web Information Systems. In: Rossi, M., Sprinkle, J., Gray, J., Tolvanen, J.-P. (eds.) Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling, DSM 2009 (2009)
4. Krahn, H., Rumpe, B., Völkel, S.: MontiCore: a Framework for Compositional Development of Domain Specific Languages. International Journal on Software Tools for Technology Transfer (STTT) 12(5), 353–372 (2010)
5. Krahn, H., Völkel, S.: Integrated Definition of Abstract and Concrete Syntax for Textual Languages. In: Engels, G., Opdyke, B., Schmidt, D.C., Weil, F. (eds.) MODELS 2007. LNCS, vol. 4735, pp. 286–300. Springer, Heidelberg (2007)
6. Rumpe, B.: Modellierung mit UML, 2nd edn. Springer (2011)
7. Rumpe, B.: Agile Modellierung mit UML: Codegenerierung, Testfälle, Refactoring. Springer (2004)

8. Schindler, M.: Eine Werkzeuginfrastruktur zur agilen Entwicklung mit der UML/P. PhD thesis, RWTH Aachen (2012)
9. Mellor, S.J., Balcer, M.: Executable UML: A Foundation for Model-Driven Architectures. Addison-Wesley Longman Publishing Co. Inc., Boston (2002)
10. Grönniger, H., Reiß, D., Rumpe, B.: Towards a Semantics of Activity Diagrams with Semantic Variation Points. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010, Part I. LNCS, vol. 6394, pp. 331–345. Springer, Heidelberg (2010)
11. Activiti Homepage, `http://activiti.org` (retreived June 30, 2012)
12. JBPM Homepage, `http://www.jboss.org/jbpm` (retreived June 30, 2012)
13. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine (2000)
14. Schwinger, W., Retschitzegger, W., Schauerhuber, A., Kappel, G., Wimmer, M., Pröll, B., Cachero Castro, C., Casteleyn, S., De Troyer, O., Fraternali, P., Garrigos, I., Garzotto, F., Ginige, A., Houben, G.-J., Koch, N., Moreno, N., Pastor, O., Paolini, P., Pelechano Ferragud, V., Rossi, G., Schwabe, D., Tisi, M., Vallecillo, A., van der Sluijs, K., Zhang, G.: A survey on web modeling approaches for ubiquitous web applications. International Journal of Web Information Systems 4(3), 234–305 (2008)
15. Brambilla, M., Ceri, S., Fraternali, P.: Process Modeling in Web Applications. ACM Transactions on Software Engineering and Methodology 15(4), 360–409 (2006)
16. Hemel, Z., Verhaaf, R., Visser, E.: WebWorkFlow: An Object-Oriented Workflow Modeling Language for Web Applications. In: Czarnecki, K., Ober, I., Bruel, J.-M., Uhl, A., Völter, M. (eds.) MODELS 2008. LNCS, vol. 5301, pp. 113–127. Springer, Heidelberg (2008)
17. Visser, E.: WebDSL: A Case Study in Domain-Specific Language Engineering. In: Lämmel, R., Visser, J., Saraiva, J. (eds.) GTTSE 2007. LNCS, vol. 5235, pp. 291–373. Springer, Heidelberg (2008)
18. Koch, N., Knapp, A., Zhang, G., Baumeister, H.: UML-based Web Engineering: An Approach based on Standards. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (eds.) Web Engineering: Modelling and Implementing Web Applications, pp. 157–192. Springer, Heidelberg (2008)
19. Kroiss, C., Koch, N., Knapp, A.: UWE4JSF: A Model-Driven Generation Approach for Web Applications. In: Gaedke, M., Grossniklaus, M., Díaz, O. (eds.) ICWE 2009. LNCS, vol. 5648, pp. 493–496. Springer, Heidelberg (2009)
20. Torres, V., Pelechano, V.: Building Business Process Driven Web Applications. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 322–337. Springer, Heidelberg (2006)
21. Rossi, G., Schmid, H., Lyardet, F.: Engineering Business Processes in Web Applications: Modeling and Navigation Issues. In: Third International Workshop on Web Oriented Software Technology, Oviedo, Spain, pp. 81–89 (2003)

Towards Conceptualizing Quality-Related Stakeholder Interactions in Software Development

Vladimir A. Shekhovtsov, Heinrich C. Mayr, and Christian Kop

Institute for Applied Informatics, Alpen-Adria-Universität Klagenfurt, Austria
shekvl@yahoo.com, {mayr,chris}@ifit.uni-klu.ac.at

Abstract. The paper addresses the issue of organizing quality-related interaction between business stakeholders and software developers relying on established common vocabulary. It establishes a conceptual representation for the process of such interaction. This conceptualization is based on a set of notions representing software quality and its particular incarnations; they are used to define the activities of the interaction process. The process is conceptualized on two levels: a coarse-grained level defining the set of generic activities and the conditions of launching these activities and a fine-grained level describing particular interaction steps in detail. The conceptualization is expected to be shared as a part of upcoming ontology of stakeholder quality perception and assessment.

1 Introduction

Software processes cannot be successful without involving the respective business stakeholders. One important category of such involvement concerns collecting stakeholder opinions and expectations on quality of the prospective system. Unfortunately, organizing such kind of involvement still remains an open problem mainly because of the difficulties in establishing the necessary communication channels between the involved parties (on the customer side and on the developer side), especially carrying quality-related information (quality-carrying channels). This difficulty can be largely attributed to the fact that these parties speak different languages while talking about quality and it usually takes a long time to agree on common vocabulary.

We propose to address this problem by establishing the intelligent support for dealing with quality-related information in the software process. It involves collecting rich descriptions of quality-related issues involving business stakeholders into a semantic repository and using this information to predict the reaction of the parties to the future issues of similar kind or to facilitate coming to the common language by these parties – with a purpose of maintaining effective quality-carrying communication channels between the customer side and the developer side. This is a goal for the ***QuASE*** project (***Qu***ality-***A***ware ***S***oftware ***E***ngineering) [19, 25] established in cooperation with two local software development companies.

The part of the above problem addressed in this paper is the issue of establishing the conceptual representation for the process of quality-related stakeholder interaction. We propose to base such representation on the common conceptualization of the quality itself which needs to be flexible enough to be able to reflect different semantics acquired

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 73–86, 2013.

by quality on different stages of this process and for different participants; we define the set of concretizations of the quality concepts reflecting the variety of these semantics.

The paper is structured as follows. Section 2 describes the case study and outlines the applied empirical methodology. Section 3 defines basic quality-related and process-related concepts and conceptualizes the stages of the interaction process. Section 4 describes the introduced concepts using a formal set-builder notation. It is followed by the description of the related work, the conclusions, and the future research directions.

2 The Case Study

We are going to base our explanation on the following case study. We investigate the software process for the IT firm (company *YY*) working with the customers in social care field to make their support systems accessible to the mobile end users (e.g. social workers). For this category of projects, customer organizations already have their own established support systems (with IT departments supporting these systems) which are made extendable by providing the interface which needs to be utilized by *YY* to develop their solution. The following research questions, among others, were stated in a process of the free-form detailed interviewing of the IT staff of the company:

1. Which participants are involved in which stages of the interaction process?
2. Is the quality understood differently on different stages of the interaction process and by different participants? If so, what are the differences?
3. Is the process represented differently for different categories of stakeholders? If so, what are the differences?
4. What is the character of the initial preconceptions on quality possessed by the customers on the start of the negotiations?
5. What could be the distance between the quality initially requested by customer and the quality specified in a contract after the negotiation?

The following conceptualization is based, in part, on the transcribed answers to these questions; to obtain this, we applied the basic techniques of *coding* and *conceptualization* stages of the grounded theory [3, 8]. We do not claim applying the complete process of grounded theory as defined e.g. in [23]; this will be the target of future research with a goal of establishing the conceptualization of the whole process of stakeholder perception of quality; for this purpose, additional cases are planned to be involved. The obtained evidence is also combined with our own experience and the knowledge incorporated into the software quality standards i.e. ISO/IEC 25010 [15].

3 The Stakeholder Interaction Process

3.1 Basic Quality-Related Definitions

We start the conceptualization of the process in question from the definition of the basic concept of quality. This definition follows [24] in relying on the body of work of formal ontology, represented by DOLCE [21] and CORE [17] ontologies.

In the subsequent description, we use underlined bold italic font with the appropriate capitalization for the concepts introduced for the first time and underlined italic font for the concepts mentioned after their introduction.

The Concept of Quality. Following and extending our earlier work [24], we define four components of the concept of quality of the SUD (software under development):

1. A set of involved ***Quality Characteristics*** (e.g. "performance" or "reliability") understood per DOLCE [21, pp. 16-18] as the categories for perceivable and measurable entities characterizing particular individuals. Every *Quality Characteristic* is accompanied by a set of ***Quality Spaces*** - conceptual spaces per Gardenfors [11]. These spaces consist of perceivable and measurable ***Quality Dimensions*** (e.g. "throughput" or "response time" for performance) with associated ***Metrics*** defining how the individuals are positioned according to the particular dimension (e.g. "the metric for calculating the throughput"). Shared *Quality Spaces* indicate a common agreement about the way of measuring *Quality Characteristic* e.g. "the response time is measured in seconds"; private *Quality Spaces* indicate the lack of such agreement e.g. "the reliability can be high or low".
2. A set of relationships among particular *Quality Characteristics* (such as those specifying their hierarchy or interdependencies).
3. A set of relationships among *Quality Characteristics* and ***Quality Subjects*** (participants of a quality-related software process activity, e.g. business stakeholders). Such relationships can e.g. define stakeholder speech acts [17, p. 181] connecting private quality spaces to stakeholders. In this, they follow CORE by categorizing ways of perceiving quality by quality subjects as a part of communicated information. Examples of speech acts are directive acts (ordering something to be done) which define precise quality constraints (if the accompanying *Quality Space* is shared) or qualitative softgoals [7] if such *Quality Space* is private.
4. A set of relationships among *Quality Characteristics* and ***Quality Objects*** (elements of a SUD decomposition the *Quality Characteristics* to be connected to, e.g. functional entities defined by a SUD architecture).

An example of the latter two relationships is the constraint "the response time (*Quality Dimension*) of the component *M* (*Quality Object*) must be below 0.5 sec according to the stakeholder *K* (*Quality Subject*)". Based on the above definitions, we introduce the ***SUD Quality*** concept as the tuple comprising all four abovementioned sets.

Quality Facets and Incarnations. Based on the above concept, we provide additional definitions necessary for our conceptualization. In a way of defining more specific views of quality such as the quality at specific stages of the software process, as seen by specific stakeholders etc., we define ***Quality Facet*** as a *SUD Quality* defined for a particular subset of *Quality Characteristics*, *Quality Subjects*, and *Quality Objects*. Also we define facet-specific set of *Quality Spaces*.

The ***Quality Point*** in *Quality Space* defines a position of the *Quality Object* according to this space's *Quality Dimensions* (e.g. "a position of the component *M* according to the throughput"). After specifying a single *Quality Point* in every *Quality Space* defined for a *Quality Facet* and taking a set of all such *Quality Points* we can define the ***Quality Incarnation*** by connecting this set to a specific software process activity and a specific purpose to serve. It concretizes the *SUD Quality* by conceptualizing particular quality levels at particular stages of the software process. An example of such incarnation can be "the (particular) performance and reliability of the component *M* perceived by the stakeholder *K* at the start of the negotiation". This concept is illustrated on Fig.1 together with the concepts it depends upon.

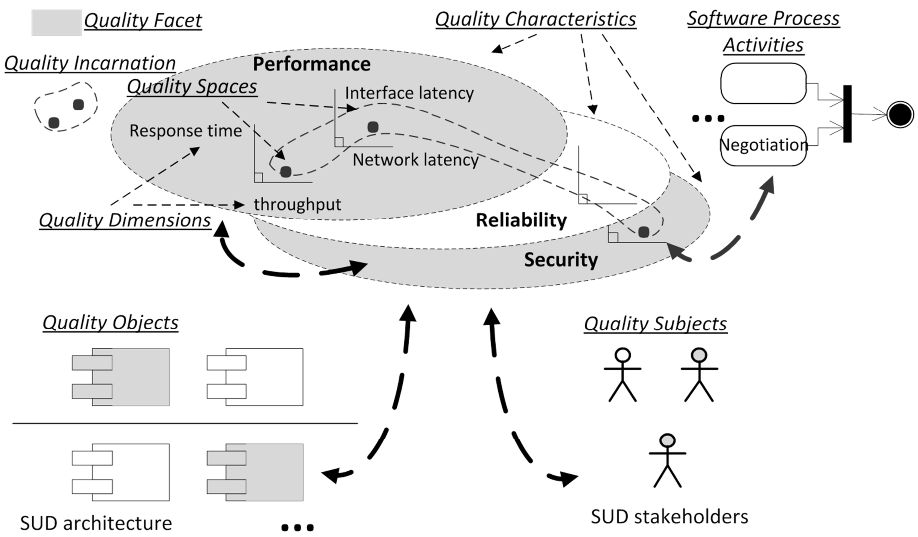


Fig. 1. Basic quality-related concepts

3.2 Basic Interaction-Related Definitions

Quality Realms. We propose to treat all the quality concepts emerging on different stages of the interaction process and all the interactions with stakeholders involving these quality concepts as belonging to specific ***Quality Realms*** defined as particular subdomains (fields of interest, concerns) in the software engineering domain uniformly influencing the treatment of the software quality for all the involved activities and participants. The quality concepts emerging at different stages of the interaction process belong to two *Quality Realms*:

1. ***User Satisfaction Realm*** for the relevant activities and participants related to the interests of the SUD stakeholders; for this realm, the quality is treated from the point of view of their satisfaction (quality in use [14, 15]);
2. ***Implementation Realm*** for the relevant activities and participants related to the interests of the IT people implementing the SUD; here, the quality is treated as the set of objective characteristics of the system (e.g. its external quality [13, 15]).

Process Participants. For our case, we define three main categories of participants:

1. ***Business Stakeholders*** do not necessary have an IT experience; they are end users of the SUD. They do not always interact with the *Developers* directly. Both their understanding of quality and their role in the quality interaction process belong to the *User Satisfaction Realm*.
2. ***Developers*** are the IT professionals responsible for developing of the software system itself. Both their understanding of quality and their role in an interaction process belong to the *Implementation Realm*.
3. ***Software Integrators*** are the IT professionals responsible for running the software platform deployed at the customer organization's site, they perform integration of the solutions provided by *Developers* into that platform. This role is twofold: on the one hand, they perceive the quality as IT professionals (i.e. as belonging to the *Implementation Realm*), on the other hand, they are the prospective customers of the SUD so their participation in quality-related stakeholder interaction belongs to the *User Satisfaction Realm*.

As we can see, there are two ***Stakeholder Roles*** which need to be defined as belonging to the particular *Quality Realm*:

1. related to the stakeholder perception of quality (distinguishing ***Business Thinkers*** and ***IT Thinkers***);
2. related to the stakeholder participation in the interaction process (distinguishing ***Business Actors*** and ***IT Actors***).

Quality realms and stakeholder roles are depicted on Fig.2.

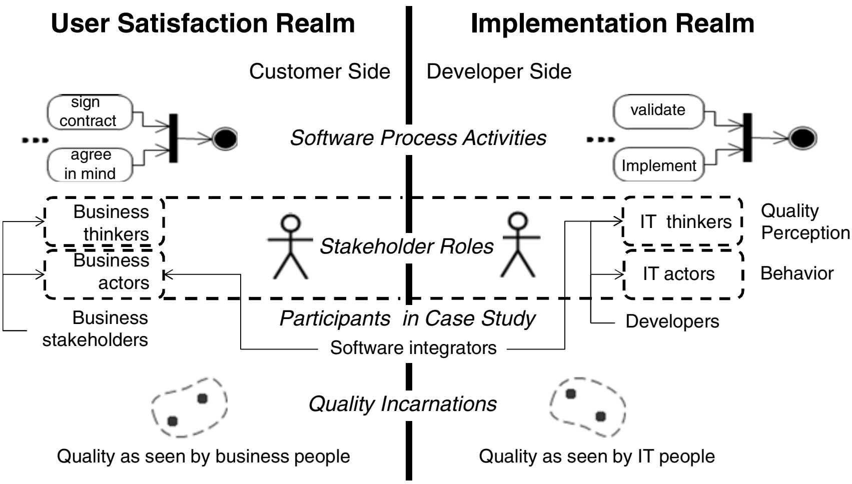


Fig. 2. Quality realms and stakeholder roles

3.3 Coarse-Grained Quality Awareness Process

Continuous Quality Awareness. Before elaborating the quality negotiation process, it is necessary to explain its place in the complete development lifecycle. To do this, we need to explain the concept of the ***Continuous Quality Awareness*** defined as the

Developers ability to check, throughout the software process, the desired *SUD Quality* (from the *Business Actor* point of view) against the current implementation stage.

To facilitate *Continuous Quality Awareness*, it is necessary to establish the set of ***Awareness Support Activities*** to be performed (triggered) continuously on different software process to provide the means of getting the correct stakeholder opinions on quality. Quality-related interaction can be considered such an activity.

Process Outline. In defining the process of interaction, we will proceed on two levels. First, we define the coarse-grained interaction process (***Awareness Support Process***). This process extends the general software process by introducing a means of triggering the more specific quality negotiation activities if necessary.

Awareness Support Process relies on the concept of the ***Interaction Trigger***. Such trigger corresponds to some event which initiates executing the set of actions for the quality-related interaction. We can distinguish three possible ways of defining such triggers:

1. Making the set of triggers correspond to the predefined points in the software process, e.g. they can be fired at its milestones; this way, the exact time of firing is known beforehand;
2. Making the triggers correspond to the predefined events in the software process, e.g. defining them as firing after making all important design decisions (e.g. "introducing new caching solution which could affect the performance"); this way, though the exact time of firing cannot be known in advance, we can know in advance the conditions of firing; these triggers are fired at the changes of the ***SUD State*** understood as the set of internal variables affecting *SUD Quality*.
3. Defining ad-hoc (or on-demand) triggers that can be fired on demand of the participating actor; this way, neither the time nor the condition can be known in advance; with such triggers, real *Continuous Quality Awareness* can be achieved (stakeholders opinions on quality can be taken into account at any time).

Now we can define the set of actions to be executed on firing the interaction trigger. We distinguish four basic coarse-grained activities:

1. ***Preparation Activity*:** the necessary preliminary actions are performed (e.g. initial version of the *SUD Quality* is produced);
2. ***Negotiation Activity*:** the sides agree to the particular incarnation of *SUD Quality* (*Pre-Implementation Negotiation*);
3. ***Implementation Activity*:** as a result of the implementation activities the SUD is transitioned to some next state possessing the particular *SUD Quality*;
4. ***Checking Activity*:** the implemented *SUD Quality* is compared against the negotiated one; if necessary, *Post-Implementation Negotiation* is performed; it is also possible to perform e.g. checking the direction of overall quality trend (indicated by e.g. the distance between the initial and agreed versions of *SUD Quality*).

We can distinguish two different sequences of coarse-grained activities depending on the type of the *Interaction Trigger*. For the triggers of the first and second kind, i.e. with firing conditions corresponding to the structure of the software process (e.g. the development milestones or design decisions) the flow of activities is depicted on

Fig.3. Here the *Checking Activity* is executed at the end of the sequence (after the *Implementation Activity*) to check if the result of the implementation achieved the agreed *SUD Quality*.

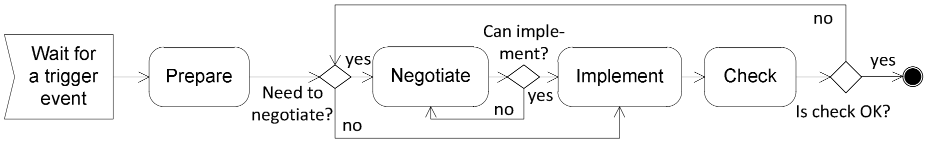


Fig. 3. Coarse-grained activities to be executed at milestone or design decision

For on-demand triggers, the sequence of activities is depicted on Fig.4. Here it is necessary to execute the *Checking Activity* on the trigger event first to check if the current *SUD Quality* corresponds to its agreed values. In this situation, both the *Negotiation Activity* and the *Implementation Activity* need to be executed only if the check was not successful.

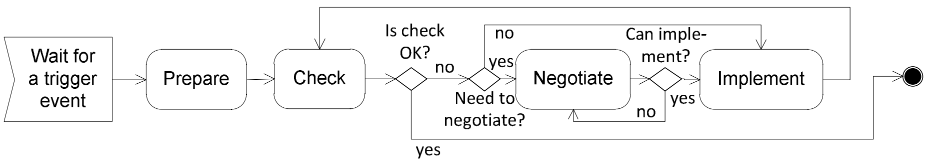


Fig. 4. Coarse-grained activities to be executed on demand

3.4 Preparation Activity and Corresponding Quality Incarnations

Before describing this activity in detail, we describe the corresponding *Quality Incarnations*. We start with those belonging to the *User Satisfaction Domain*. The informal schema of the fine-grained activities and the corresponding *Quality Incarnations* is depicted on Fig.5.

Expected Quality and Its Boundaries. The ***Expected Quality*** can be defined as the *Quality Incarnation* representing an image of the *SUD Quality* that is conceived of by the individual *Business Actor* as a negotiable quality he/she wants the final SUD version to provide. Such *Quality Incarnation* needs to be taken into account every time when the software process deals with *Business Actors*' opinions, but it exists only in a *Business Actor*'s mind (e.g. "now I (the stakeholder *K*) think that the response time as seen through the user interface (UI) element *L* may not exceed 0.5 sec".) It can change throughout the interaction process as a result of the evolution of *Business Actor*'s quality vision. It belongs to different *Quality Realms* for the interactions with *Business Thinkers* and *IT Thinkers*.

Expected Quality Boundaries are the *Quality Incarnations* reflecting the fact that the *Business Actor* enters the process of interaction with specific quality-related preconceptions in mind constituting the *Expected Quality* permissible range; it depends on the *Business Actor*'s capabilities to accept the *SUD Quality*. The ***Best-Case***

Quality is the higher bound of the *Expected Quality*: if the *SUD Quality* as perceived by *Business Actor* equals or exceeds the *Best-Case Quality*, the particular SUD version can be immediately accepted as completely satisfying the *Business Actor*; no additional interactions are necessary. The ***Last-Resort Quality*** is the lower bound of the *Expected Quality*: if the *SUD Quality* is below the *Last-Resort Quality* boundary, the SUD version can be immediately rejected as dissatisfying the *Business Actor*: e.g. "if I (the stakeholder *K*) will get the response time seen through the UI element *L* of above 15 sec I will think twice about dealing with these developers again".

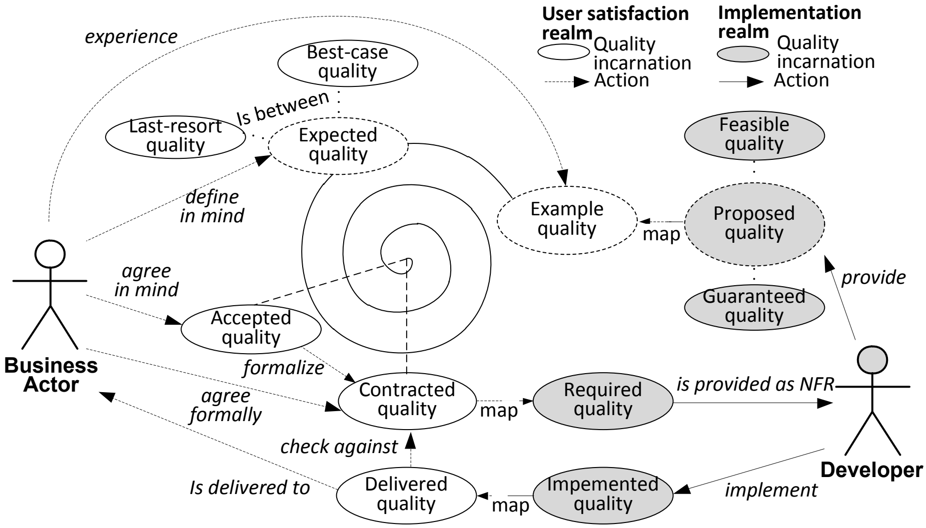


Fig. 5. Fine-grained interaction activities and corresponding quality incarnations

Proposed and Example Quality and Their Boundaries. We define our conceptualization for the case when the interaction with *Business Actor* is performed from the *Developer* side by making the *Business Actor* experience some version of quality which could be then the subject of negotiation: a ***Proposed Quality***. It denotes a set of values for quantified quality attributes of the SUD reflecting the current state of its development. The SUD is expected to have that quality if implemented exactly as defined at the given development stage. *Proposed Quality* can be produced based on a (e.g. simulation) model, a system prototype, or the complete system version. An example could be "the vertical prototype (or the performance simulation) of the component *M* shows the response time of 0.34 sec in the context *T*").

From the *Business Thinker* side, it is difficult to experience and assess the *Proposed Quality* directly as it is just a set of (objective) values which reflects the current state of the SUD (*external quality* in terms of [13, 15]), so it belongs to the *Implementation Realm* (e.g. "the component *M* is the element of the SUD architecture not directly visible to the stakeholder *K*"). Following [22], we propose to map this *Quality Incarnation* into the one belonging to the *User Satisfaction Realm*: an ***Example Quality***. Its values need to reflect the quality in use according to e.g. ISO/IEC 9126-4 [14] and the future version of ISO/IEC 25024; they can be experienced by *Business*

Thinker naturally (e.g. "map the response time of the component *M* into the response time as shown through the UI element *L* which is understandable by the stakeholder *K*"). This mapping is not necessary in case of interacting with *IT Thinkers*.

Proposed Quality boundaries are the *Quality Incarnations* reflecting the range of the *Developer* capabilities to produce the SUD with the particular level of quality. The higher bound (***Feasible Quality***) reflect the best quality that can be produced given the available resources, whereas the lower bound (***Guaranteed Quality***) reflects the level of skills and self-esteem of *Developers* which makes producing the SUD with lower quality unacceptable for them.

Preparation Activity. Launching this activity entails executing the following fine-grained activities necessary for the preparation of the participants to the quality-related stakeholder interaction.

1. ***Establish Expected Quality Boundaries***: *Business Stakeholder* establishes in his/her mind both *Best-Case Quality* and *Last-Resort Quality*.
2. ***Establish Proposed Quality Boundaries***: *Developer* learns about or establishes both *Feasible Quality* and *Guaranteed Quality*;
3. ***Produce Initial Proposed Quality***: *Developer* produces some version of *Proposed Quality* (through e.g. SUD prototype) falling between its boundaries; it is then mapped into *Example Quality* if necessary; to be concise, below we refer to such quality (either mapped or unmapped) as *Example Quality.*

3.5 Negotiation Activity

Executing the *Negotiation Activity* entails executing the set of fine-grained activities:

1. ***Experience Proposed Quality*****:** the interaction begins from the *Developer* making *Business Actor* experience the produced *Example Quality*, after this, *Business Actor* compares it in his/her mind to the *Expected Quality* boundaries. As a result, two mutually exclusive activities can be executed:

   – ***Accept the SUD State*****:** If the *Example Quality* exceeds the *Best-Case Quality*, the SUD state possessing the particular *Example Quality* is accepted without further interactions; such experience can be of the great service to the project as it establishes the feeling of trust between *Business Actors* and *Developers*.
   – ***Reject the SUD State*****:** if the *Example Quality* falls below the *Last-Resort Quality*, the particular version of the SUD is rejected without the possibility of further interaction; it can be damaging to the whole project.

2. ***Negotiate Specific Quality*****:** If the *Example Quality* falls into the acceptable *Expected Quality* range, the process follows the ***Negotiation Iterations***; on every such iteration, the following activity is performed:

   – ***Adjust the Proposed Quality* and *Expected Quality*****:** *Developer* makes *Business Actor* experience the new version of *Example Quality* and *Business Actor* assesses his/her experience again; in doing this, every side tries to insist on some preferable quality values: in particular, *Developer* tries to keep the *Example Quality* closer to the *Guaranteed Quality* to spend fewer resources, whereas *Business Actor* tries to keep the *Expected Quality* closer to the *Best-Case Quality* as it is better correspond to what he/she has in mind initially. Throughout this iterative process both sides learn about the capabilities of each other: as a result, their *Quality Incarnations* converge to the agreement point. It could look like: “the developers show the stakeholder *K* (through the UI element *L*) the vertical prototype of the component *M* with a (UI-mapped) response time of 0.7 sec, *K* does not quite like it, so they produce a new version with a response time of 0.5 sec and *K* agrees on that”.

3. ***Agree Upon Quality*:** the negotiation process eventually ends with agreeing upon a version of the *Example Quality* shown to the *Business Actor*: the ***Accepted Quality***.

The above activities are defined for the case when the negotiation between *Business Actor* and *Developer* takes place through a sequence of *Example Quality* incarnations shown to *Business Actor* and an evolution of the *Expected Quality* incarnations in *Business Actor*’s mind; in reality, simpler sets of activities could be performed (e.g. missing the second step so the initial *Example Quality* is either immediately accepted or immediately rejected). Main concepts related to this activity are shown on Fig.6.

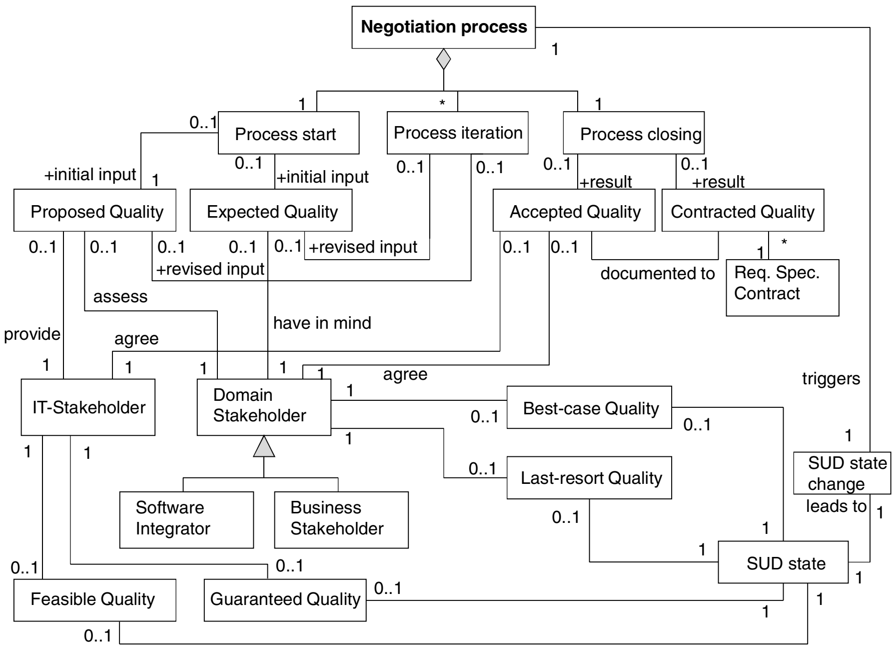


Fig. 6. Main concepts of the stakeholder negotiation activity

3.6 Implementation and Checking Activities

Contracted and Required Quality. As a result of the iterative interactions, the sides agree on *Accepted Quality*. Before communicating it to the *Developer*, it is necessary to transfer it into the official, documented form: a ***Contracted Quality*** (e.g. “the contract states that the response time seen through the user interface element *L* must not exceed 0.5 sec"). For interacting with *Business Stakeholders*, it needs to belong to the *User Satisfaction Realm*, this is not necessary for the interaction with *IT Thinkers*.

In case of *Business Stakeholder*-involving interaction, the *Contracted Quality* cannot be directly used to guide the software development activities to be performed by *Developers*. To do so, it needs to be mapped into ***Required Quality*** belonging to the *Implementation Realm*, it can be seen as the quality expressed in the requirement specification contract as a set of SUD non-functional requirements (NFR); so this mapping can be seen as a part of the NFR elicitation process (e.g. **"**the NFR *R* states that the response time of the component *M* must not exceed 0.5 sec**"**). For the interaction with *IT Thinkers*, *Contracted Quality* can be directly used as *Required Quality*.

Implementation Activity and Related Quality Incarnations. After the *Required Quality* was defined and made available to the *Developers*, the necessary implementation-related actions are performed as a part of coarse-grained *Implementation Activity*. As a result, the SUD transitions itself into the new development-related state (corresponding to e.g. the milestone). We refer to this state as possessing the ***Implemented Quality*** belonging to the *Implementation Realm*; in case of interaction with *Business Stakeholders* it needs to be mapped into the *User Satisfaction Realm* to obtain the ***Delivered Quality*** (e.g. "we delivered the SUD version with the response time of 0.45 sec as seen through the UI element *L*").

Checking Activity. After the *Implementation Activity* is completed (or right on a trigger event in case of asynchronous launching of the interaction), the obtained *Delivered Quality* needs to be experienced by the *Business Actors* as a part of the *Checking Activity*. Here, various checks can be performed, the most common one is checking if *Delivered Quality* corresponds to the *Contracted Quality*. If this is the case, next round of the *Awareness Support Process* can be started by executing the *Preparation Activity* aligned with the next ***SUD State Change*** (e.g. with the process stage bounded by the next milestone), otherwise it corresponds to the major development problem with a need of emergency actions.

4 Formal Description

In this section, we use set-builder notation to describe the introduced concepts. A *SUD Quality* is defined as a tuple $Q(C,U,F)=\left\langle C,R_q(C),R_s(C,U),R_f(C,F)\right\rangle$, where $C=\left\{c=\left\langle P(c),S(c)\right\rangle\right\}$ is a set of involved *Quality Characteristics* (comprised of a set of properties $P(c)$ and a set of *Quality Spaces* $S(c)$), $R_q(C)=\left\{r_q\left(C'\subseteq C\right)\right\}$ is a set of relationships among *Quality Characteristics* (*C'* is a set of involved characteristics, *C* is the set of available characteristics), $R_s(C,U)=\left\{r_s\left(C'\subseteq C,U'\subseteq U\right)\right\}$ is a set of relationships among *Quality Characteristics* and *Quality Subjects* (*U'* is a set of involved subjects, *U* is the set of available subjects), $R_f(C,F)=\left\{r_f\left(C'\subseteq C,F'\subseteq F\right)\right\}$ is a set of relationships among Quality Characteristics and Quality Objects (F' is a set of involved objects, F is the set of available objects).

A Quality Facet is defined as $\bar{Q} = Q\left(\bar{C} \subseteq \mathbf{C}, \bar{U} \subseteq U_{BS} \cup U_{SI} \cup U_{ITS}, \bar{F} \subseteq \mathbf{F}\right)$, where $\mathbf{C}$ and $\mathbf{F}$ are the sets of existing Quality Characteristics and Quality Objects, U_{BS}, U_{SI}, U_{ITS} are the sets of participants belonging to the particular categories. In addition, $\bar{S}(\bar{Q}) = \left\{ \bigcup S(c) \,\middle|\, c = \langle P(c), S(c) \rangle \in \bar{C} \right\}$ defines a facet-specific set of Quality Spaces.

A Quality Incarnation for the interaction process activity a is defined as a tuple $I(\bar{Q}) = \left\langle P(I(\bar{Q})), \left\{ v \,\middle|\, s(v) \in \bar{S}(\bar{Q}) \right\}, a \in A \right\rangle$, where $P(I(\bar{Q}))$ is the set of properties, A is a set of defined process activities, $s(v)$ is a Quality Space for a Quality Point v.

5 Related Work

Our research is related to both quality conceptualization and software process conceptualization so we need to consider the state of the art in these two research fields.

We published a detailed literature review of the available quality conceptualization techniques in [24], so we refer to this paper for the complete treatment of this issue. Among these techniques, it is possible to distinguish model-based [6, 9, 10, 27, 28] and ontology-based [16-18, 20] approaches, for the purpose of our research we are mostly interested in the latter. We found that these techniques are not completely suitable for our problem as they mostly concerned with conceptualizing the quality itself. They, as a rule, do not relate the proposed conceptualizations to the software process activities as seen from the both sides of the development process while performing the quality-related negotiation (e.g. they do not include the concept of quality realm or conceptualize the inter-realm mappings, also, the negotiation support is limited).

Software process conceptualization techniques [1, 2, 4, 26], on the other hand, often miss appropriate quality treatment: they do not include the notions of quality related to different software process activities. Among these techniques, we distinguish the work by Adolph and Kruchten [4] where they propose the conceptualization of the software process centering on the notion of the negotiation between the sides of the development process with a purpose of reaching common understanding; being the high-level conceptualization, it does not include the specifics of the negotiation process related to quality-carrying communication. We can see our research as an extension to that work focusing on establishing quality-carrying communication channels.

6 Conclusions and Future Work

In this paper, we established the conceptualization for the process of quality-related stakeholder interaction in software development. As basic notions, it provides the concept of SUD Quality (based on the notions of formal ontology) together with the conceptual notions of *Process Participant*, Quality Realm, and the set of quality values (Quality Incarnation). The proposed process conceptualization is based on the notion of *Continuous Quality Awareness* defining the set of interaction activities as an extension of the generic software process making possible on-demand stakeholder involvement with a purpose of checking the current *SUD State*. It defines the necessary process activities, the relevant workflow, and *Quality Incarnations* related to these activities. Its advantages are as follows:

1. It establishes a common ground for organizing the knowledge about the specifics of establishing quality-carrying communication channels into the semantic repository – a knowledge base. This information could be reused directly while organizing future interactions; it also could be used to predict the behavior of the parties while encountering similar quality-related issues;
2. It facilitates coming to a common language by different parties in the software process by providing the set of common quality-related concepts that could be utilized by all these parties while participating in quality-related interaction activities;
3. It can help in establishing process-oriented support solutions facilitating quality-related stakeholder involvement into the software process.

In future, in a framework of the QuASE project [19, 25], we plan to integrate the proposed conceptualization of the interaction process into a common ontology (*QuOntology*) which has to incorporate the knowledge about software quality and its perception by business stakeholders and developers, the processes of stakeholder quality assessment, the ways of producing quality to be proposed to stakeholders and other knowledge related to this domain. After establishing QuOntology, following the Ontology-based Software Engineering paradigm [5, 12], we plan to make it play a central role in implementing the tool support for the stakeholder involvement into the software process by e.g. facilitating semantic-based reuse of the information about quality-related interactions involving business stakeholders or the prediction of the stakeholder behaviour for the future interactions of this kind.

References

1. Acuna, S.T., Juristo, N. (eds.): Software Process Modeling. Springer, Heidelberg (2005)
2. Acuna, S.T., Sanchez-Segura, M.I. (eds.): New Trends in Software Process Modeling. World Scientific, New York (2006)
3. Adolph, S., Hall, W., Kruchten, P.: Using grounded theory to study the experience of software development. Empirical Software Engineering 16, 487–513 (2011)
4. Adolph, S., Kruchten, P.: Reconciling Perspectives: How People Manage the Process of Software Development. In: AGILE 2011, pp. 48–56. IEEE, New York (2011)
5. Bachmann, A., Hesse, W., Ru, A., Kop, C., Mayr, H.C., Vohringer, J.: A Practical Approach to Ontology-based Software Engineering. In: EMISA 2007. LNI, vol. P-119, pp. 129–142. GI, Bonn (2007)
6. Carvallo, J.P.: Systematic Construction of Quality Models for COTS-Based Systems. PhD Thesis. Universitat Politècnica de Catalunya, Barcelona (2005)
7. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Boston (1999)
8. Coleman, G., O'Connor, R.: Investigating software process in practice: A grounded theory perspective. The Journal of Systems and Software 81, 772–784 (2008)

9. Cote, M., Suryn, W., Georgiadou, E.: Software Quality Model Requirements for Software Quality Engineering. In: 14th International Conference on Software Quality Management, Quebec, pp. 31–50 (2006)
10. Deissenboeck, F., Juergens, E., Lochmann, K., Wagner, S.: Software quality models: Purposes, usage scenarios and requirements. In: WoSQ 2009, pp. 9–14. IEEE, New York (2009)
11. Gärdenfors, P.: Conceptual Spaces: A Geometry of Thought. MIT Press, Cambridge (2000)
12. Hesse, W.: Ontologies in the Software Engineering process. In: Proc. EAI 2005. Ceur-WS.org, vol. 141, pp. 3–16 (2005)
13. ISO/IEC 9126-2:2003: Software Engineering – Product Quality – Part 2: External Metrics. ISO (2003)
14. ISO/IEC 9126-4:2004: Software Engineering – Product Quality – Part 4: Quality-in-Use Metrics. ISO (2004)
15. ISO/IEC 25010:2011: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. ISO (2011)
16. Jureta, I.: Essays in Information Management: Contributions to the Modeling and Analysis of Quality in Information Systems Engineering, PhD Thesis. Department of Business Administration. University of Namur (2008)
17. Jureta, I., Mylopoulos, J., Faulkner, S.: A core ontology for requirements. Applied Ontology 4, 169–244 (2009)
18. Kabilan, V., Johannesson, P., Ruohomaa, S., Moen, P., Herrmann, A., Ehlfeldt, R.M., Weigand, H.: Introducing the Common Non-Functional Ontology. In: Gonçalves, R.J., Müller, J.P. (eds.) Enterprise Interoperability II, pp. 633–645. Springer, Heidelberg (2007)
19. Kaschek, R., Kop, C., Shekhovtsov, V.A., Mayr, H.C.: Towards Simulation-Based Quality Requirements Elicitation: A Position Paper. In: Rolland, C. (ed.) REFSQ 2008. LNCS, vol. 5025, pp. 135–140. Springer, Heidelberg (2008)
20. Masolo, C., Borgo, S.: Qualities in formal ontology. In: Foundational Aspects of Ontologies Workshop at KI 2005, Koblenz, pp. 2–16 (2005)
21. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: The WonderWeb Library of Foundational Ontologies. WonderWeb Deliverable D18. Ontology Library (final). . ISTC-CNR, Trento (2003)
22. Raubal, M.: Formalizing Conceptual Spaces. In: Proceedings of Formal Ontology in Information Systems, FOIS 2004, pp. 153–164. IOS Press, Amsterdam (2004)
23. Sharmaz, K.: Constructing Grounded Theory. Sage Publications, Thousand Oaks (2006)
24. Shekhovtsov, V.A.: On the evolution of quality conceptualization techniques. In: Kaschek, R., Delcambre, L. (eds.) The Evolution of Conceptual Modeling. LNCS, vol. 6520, pp. 117–136. Springer, Heidelberg (2011)
25. Shekhovtsov, V.A., Mayr, H.C., Kop, C.: Stakeholder Involvement into Quality Definition and Evaluation for Service-Oriented Systems. Accepted for Publication in Proc. ICSE 2012 Workshops. IEEE, New York (2012)
26. Software Process Engineering Metamodel (SPEM) 2.0. OMG (2008)
27. Tian, J.: Quality-Evaluation Models and Measurements. IEEE Software 21, 84–91 (2004)
28. Wagner, S., Lochmann, K., Winter, S., Goeb, A., Klaes, M.: Quality models in practice: A preliminary analysis. In: ESEM 2009, pp. 464–467. IEEE, New York (2009)

Agent-Based Implementation for the Discovery of Structural Difference in OWL-DL Ontologies

Maxim Davidovsky[1], Vadim Ermolayev[2], and Vyacheslav Tolok[3]

[1] Zaporozhye National University, Center of Information Technologies, Zhukovskogo Str. 66, 69600 Zaporozhye, Ukraine
[2] Zaporozhye National University, Department of Information Technologies, Zhukovskogo Str. 66, 69600 Zaporozhye, Ukraine
[3] Zaporozhye National University, Department of Mathematical Modelling, Zhukovskogo Str. 66, 69600 Zaporozhye, Ukraine
m.davidovsky@gmail.com, vadim@ermolayev.com, vyacheslav-tolok@yandex.ru

Abstract. Various knowledge-based information systems contain distinct knowledge representations reflecting different domains of interest and different viewpoints across domains of discourse. For efficient use of knowledge-based systems it is necessary to know semantic relations or alignment between different knowledge representations. One of the promising approaches is the use of intelligent software agents where agents communicate in order to align respective knowledge representations. The paper presents an approach for ontology alignment based on implementation of meaning negotiation between intelligent agents. In the approach, negotiation is conducted in iterative way. At each step agents compare ontological contexts and use propositional substitutions in order to reduce semantic distance between the contexts. The focus of the paper is the implementation of agents' negotiation strategy.

Keywords: Ontology, ontology alignment, intelligent agent, meaning negotiation, implementation.

1 Introduction

Complexity and dynamism of modeling domains lead to heterogeneous and polysemous knowledge models and representations. Hence, knowledge-based information systems inevitably face with knowledge heterogeneity problem. Today the mainstream of knowledge representation is ontologies. Thus, for efficient use of knowledge-based systems it is necessary to know semantic relations between knowledge representations or alignment between respective ontologies. A perspective approach for ontology alignment is the one based on the use of intelligent software agents.

The approach is promising both from the efficiency standpoint and in terms of appropriateness for a broad range of applications. Today, ontologies are widely used for knowledge management, information retrieval and sharing, e-commerce, and

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 87–95, 2013.

many other applications. However, ontologies are often used in the open, decentralized and distributed systems where different nodes possess distinct though semantically overlapping ontologies. Also, ontologies can evolve independently of each other. Thus, for the interaction between parts of a distributed system and to ensure proper operation of the entire system the harmonization of ontologies must be implemented. In such circumstances, the alignment should be carried out in automatic mode. Moreover, the alignment should be performed dynamically in response to changes of the system, or "on demand" – during the interaction of previously non-interacting parties or those which do not have permanent relation or contact. The capability of migration of system nodes according to the structure changes over time is one more important aspect. It is obvious that the above requirements demand some intelligence from the parties involved in the alignment. Moreover, it must be “distributed”, “mobile” and “dynamic” intelligence. To the authors' opinion, such direction of research within the bounds of distributed artificial intelligence as the agent paradigm is one of the most consistent with and appropriate to the aforementioned requirements.

Another topical issue is the use of ontologies in such an open and decentralized system as the Semantic Web (SW) [1]. Ontologies are one of the pillars of the SW where they are used as the structural frameworks for knowledge representation and organizing of information. Another important component of the SW is the use of intelligent software agents which also makes the use of the agent paradigm the natural and organic means for aligning ontologies in the SW. Thus, intelligent agents and Semantic Web services that serve as a functional framework of the SW can interact seamlessly by aligning respective ontologies or asking for alignment services to the third party agent(s). A more detailed analysis of the use cases and respective specific requirements for ontology alignment is presented in our recent paper [2] as well as several known to date agent-based solutions.

In this paper we present an approach for ontology alignment based on implementation of meaning negotiation between intelligent agents. The negotiation strategy implies aligning ontologies by parts (conceptual subgraphs or contexts) that are relevant to a particular negotiation encounter. Negotiation is conducted in an iterative manner and is aimed at the reduction of a semantic distance between the contexts. Agents use propositional substitutions which may reduce the distance and support them with argumentation. The process is stopped when the distance reaches some commonly accepted threshold or the parties exhaust their propositions and arguments.

The paper is organized in the following way. In section 2 we describe the entire alignment process by using a characteristic task of ontology instance migration. Section 3 is devoted to implementation details. Section 4 outlines the set-up for future evaluation experiments. And the 5th section concludes the paper with a brief description of intermediate results and our plans for future work.

2 Ontology Alignment Process

Ontology alignment process (referred also as ontology matching [3]) is a process of discovering the correspondences (or mappings) between the elements of different ontologies. In the presented agent-based implementation the process embodies the strategy of automated meaning negotiation proposed by Ermolayev et al. in [4] for efficient information retrieval in open and distributed environments. The presented work is a part of the research activity aiming at development and implementation of an efficient methodology of ontology alignment and instance migration in decentralized settings (see also [5] for details on instance migration part of the methodology). The methodology (depicted on figure 1) assumes (semi-) automated iterative process of ontology alignment and instance migration with possible human intervention for checking the correctness and setting up the process. The presented implementation realizes the first necessary step of the methodology – discovery of mappings between respective ontological elements and presentation of results (r.t. TaskKind shapes No I and II on the figure) in the form suitable for further (automatic) processing and (semi-automated or manual) checking and correction.

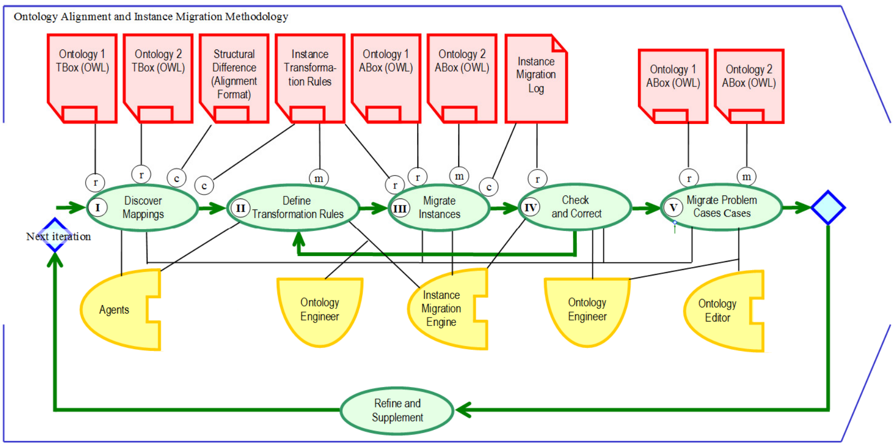


Fig. 1. Ontology alignment and instance migration methodology specified in ISO/IEC 24744 notation for describing methodologies [6]

Let's consider the task of ontology instance migration – a typical and characteristic case for knowledge sharing and reuse scenarios. Suppose we have an ontology O_1 conceptualizing knowledge of a domain P. In general, an ontology, as a rule, contains terminological part (or TBox) and assertional part (or ABox). TBox represents conceptual knowledge of a domain. ABox is a set of facts about the domain or, in other words, the instances of respective concepts and concrete values of their properties. Hence, we can represent the ontology O_1 as a tuple $O_1 = \langle C_1, I_1 \rangle$, where C_1 is a set of classes and respective properties representing concepts and relationships between them. I_1 is a set of instances.

Assume we have some ontology $O_2 = \langle C_2, I_2 \rangle$ reflecting some other perspective of the domain P or modeling some other domain $\tilde{P}$ semantically overlapping with P. Thus, O_2 contains the sets $C_2 \neq C_1 \neq \emptyset$ and $I_2 \neq I_1$. Obviously, for more complete and efficient use of knowledge we have to enrich the ontology O_2 with assertional knowledge of O_1 (i.e., some set of instances $\widetilde{I_1} \subseteq I_1$) that is relevant and valid for O_2.

We also assume that the ontologies O_1 and O_2 exist in decentralized settings and are associated with some operational agents Ag_1 and Ag_2 respectively. Then the process of replenishing knowledge of the ontology O_2 consists of two steps: (1) matching of terminological parts C_1 and C_2 by the agents and obtain some structural difference ΔC; and (2) being based on the structural delta migrate the set of instances $\widetilde{I_1}$, and obtaining a new set of instances $\widetilde{I_2} = I_2 \cup \widetilde{I_1}$ for ontology O_2.

Consider the first step of the process (Fig. 2). At the first step the agent Ag_2 initiates the process of ontology alignment and starts negotiation with the agent Ag_1. Then Ag_2 iterates over all concepts $c2_n \in C_2$ along with the respective contexts $Ctx2_k$ (where $n = \overline{1, N}$, $k = \overline{1, K}$, N and K are the numbers of concepts and contexts in O_2 respectively). Here, we define a context of concept c as a tuple $\langle \widehat{C}, \hat{I} \rangle$, where $\hat{C}$ is a set of TBox elements directly related with c, and $\hat{I}$ is a set of respective ABox elements. Then Ag_2 sends the contexts to the agent Ag_1. Agent Ag_1 applies the obtained context to the ontology O_1 and forms hypotheses of equivalence of the obtained concept and concepts of the ontology O_1. The hypotheses are weighted with confidence ratios on the basis of the measurement of contexts' semantic distances SD, which are calculated as sums of the semantic distances between the respective contexts' constituents. To calculate the semantic differences SD the variety of metrics can be used (e.g., see an overview of such metrics in [7]). In our implementation SD is implemented as an interface (see Section 3 Implementation) that allows substituting different concrete implementations of SD or their combination for more efficient discovering of correspondences.

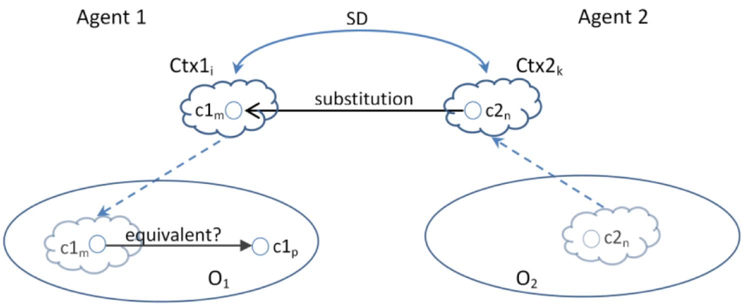


Fig. 2. Illustration of the agent-based ontology alignment process

To test the approach we used the following metrics:

1. Lexical metric. Let R_A, R_B be the sets of roots of the words which constitute the names of concepts A and B respectively, then lexical similarity may be computed as:

$$Sim_L = \frac{R_A \cap R_B}{R_A \cup R_B} \tag{1}$$

2. Instance similarity metric. Two concepts, A and B are similar if $I_A \cap I_B \neq \emptyset$, where I_A and I_B are the sets of instances of A and B respectively. Instance Similarity is measured by symmetric Jaccard coefficient [8]:

$$Sim_I(A,B) = \frac{P(I_A \cap I_B)}{P(I_A \cup I_B)} \tag{2}$$

where $P(I)$ is the probability that a randomly chosen instance belongs to I.

3. Contextual or Feature Similarity. Similarity between feature sets may be computed by Tversky metrics [9]. The set of similarity measures of the object properties and related concepts s_j used as the feature set. Hence, feature similarity may be computed as an integrative metric for a pair of concepts as follows:

$$Sim_C = \frac{1}{m}\sum_{j=1}^{m} s_j \tag{3}$$

The particular negotiation round stops when the agents exhaust their argumentation and all of the respective contexts' constituents have been used. The whole process stops when there are no more contexts for all agents according to the alignment task.

Further, agents produce output of the resulting alignment of the two kinds (Fig. 3). The first type of result is a representation of the alignment in the Alignment format[1] [10]. This representation is quite intuitive and allows to conveniently assess the correctness and completeness of the resulting alignment. Also, this format is the de facto standard for representing alignments for participants of OAEI[2] and the presentation of results in this format allows us to estimate the system using OAEI benchmarks and makes the results comparable to those provided by OAEI. The second type of output is the representation of the alignment in the form of structural

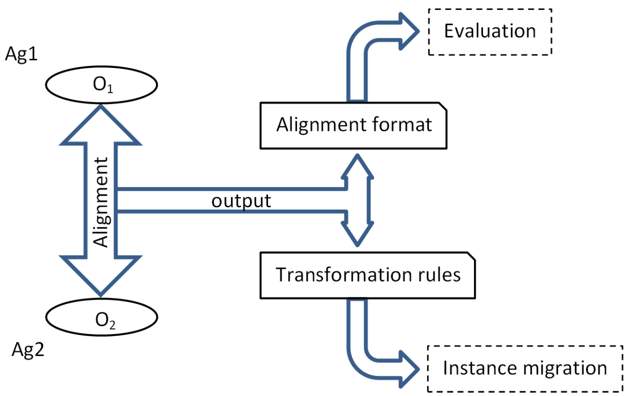


Fig. 3. Two kinds of alignment representation

1 http://alignapi.gforge.inria.fr/format.html
2 http://oaei.ontologymatching.org/

difference ΔC and generation of respective transformation rules for the next step – the migration of ontology instances (the problem statement of ontology instance migration and the description of solution based on transformation patterns and rules are described in detail in [5]).

At the second step agents use Instance Migration Engine in order to transfer instances between ontologies O_1 and O_2 based on the transformation rules generated at the first step. Problem cases occurring during the migration are recorded into the migration log (possible problems and collisions during the migration are considered and analyzed in [11] and [5]). Finally, the target ontology agent asks a reasoner for the consistency check of the obtained ontology O_2. Entities that break the consistency are not added and are written to the log. All problem cases can be analyzed and resolved in subsequent iterations (if any) of the ontology alignment process.

3 Implementation Details

To implement the described ontology alignment process we have developed a system of two intelligent agents. We use JADE[3] (Java Agent Development Environment) as an agent platform. Each agent contains an ontology processing module – ontology handler, which allows to parse an ontology and iterate over ontological elements and a context handler for the management of ontological contexts. Structural delta between the contexts is determined by the use of special determinator and computed based on various implementations of semantic distance metrics. The Agents carry out the alignment of ontologies presented in the OWL[4] language and use the OWL API[5] for coping with OWL-ontologies. In Fig. 4 is depicted a UML diagram fragment of the proof-of-concept software (please note that the figure shows not all the classes and methods in details, but the functional skeleton of classes and interfaces for a better understanding of the relationships between them).

The Ontology Alignment API [10] has been used for representing alignments in the Alignment format. The API provides a set of interfaces that can be implemented using different methods and metrics. A typical example of such an interface is the *AlignmentProcess* interface that provides an abstract method *align()* which can be implemented by special ontology matching techniques. In our implementation, the agents exchange ontological contexts rather than the whole ontologies, and the method implements alignment of individual contexts. The resulting alignment is obtained by combining the alignments obtained in separate negotiation iterations and applying the cut-off *trim()* operation. This operation cuts off the alignment in accordance with a predetermined threshold, removing the elementary alignments with confidence values below the threshold. As noted above, on the basis of the obtained alignment an agent forms the set of transformation rules to convert and transfer the

[3] `http://jade.tilab.com/`
[4] `http://www.w3.org/TR/owl2-overview/`
[5] `http://owlapi.sourceforge.net/`

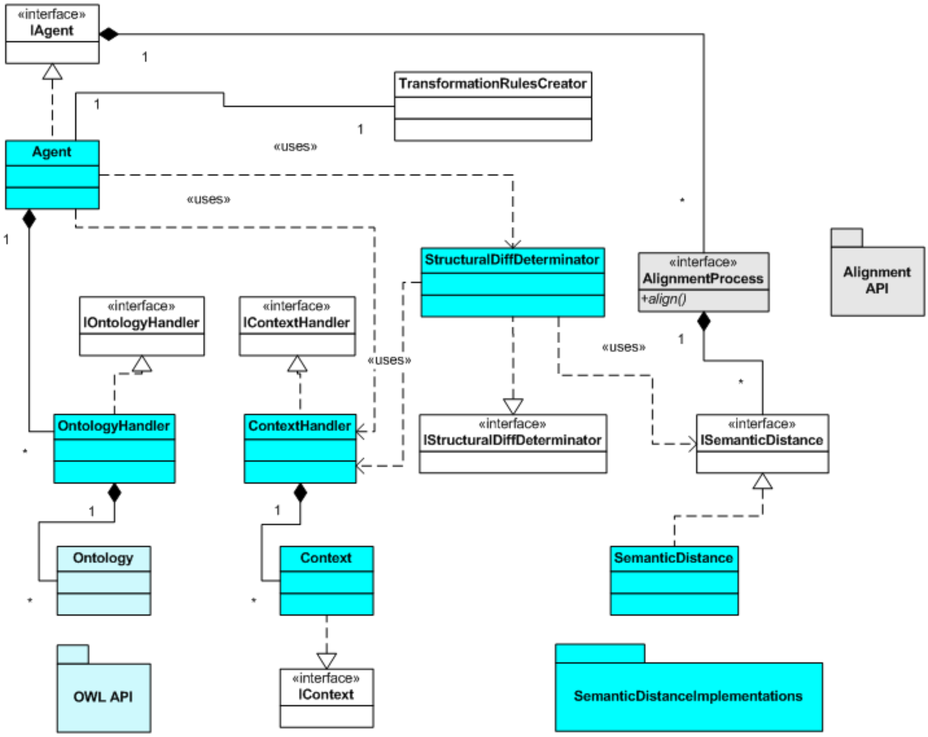


Fig. 4. An implementation UML diagram fragment

ontology instances. More information about the transformation pattern and rule language can be found in [5]. For consistency checking in our implementation we use the Pellet reasoner[6].

4 Evaluation Set-Up

The implemented software is planned to be evaluated for quality and completeness of the resulted alignments. The experimental set-up is depicted in Fig. 4. According to the two-step character of the alignment process we can measure the quality after each step on separate iterations. For these purposes we plan to use *Precision*, *Recall* and *F-measure* metrics (the description of these metrics for measuring the quality and completeness of ontology instance migration is given in [5]; the use of the metrics for assess ontology alignments is quite similar). We plan to use OAEI ontologies and datasets as a test-bed for our experiments.

[6] `http://clarkparsia.com/pellet/`

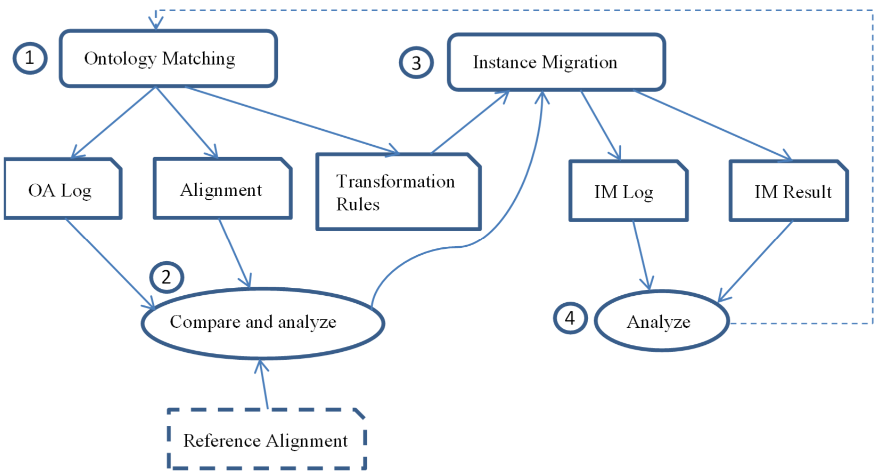


Fig. 5. Set-up for ontology alignment evaluation experiment (used abbrs.: OA – Ontology Alignment, IM – Instance Migration)

After the first step we evaluate the obtained alignments by comparing them with the reference ones. Then we proceed with instance migration and consequently analyze the obtained results. Being based on the analysis we can refine the set of alignments. Thus, we can assess the effectiveness of separate software components (used for matching and instance migration subtasks) as well as the iterative methodology in general.

5 Conclusions and Future Work

The paper presents the implementation of agent-based ontology alignment. The implementation is a continuation of work on realization of the iterative ontology alignment and instance migration methodology with the use of intelligent software agents. This methodology provides an apparatus for solving knowledge heterogeneity problem in decentralized environment and addresses a number of problems related to knowledge sharing and reuse. In the future we plan to conduct series of experiments with the developed software.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284, 28–37 (2001)
2. Davidovsky, M., Ermolayev, V., Tolok, V.: A Survey on Agent-Based Ontology Alignment. In: Proc. Int. Conf. on Agents and Artificial Intelligence (ICAART 2012), Vilamoura, Algarve, Portugal, February 6-8, pp. 355–361 (2012)
3. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)

4. Ermolayev, V., Keberle, N., Matzke, W.-E., Vladimirov, V.: A Strategy for Automated Meaning Negotiation in Distributed Information Retrieval. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 201–215. Springer, Heidelberg (2005)
5. Davidovsky, M., Ermolayev, V., Tolok, V.: Instance Migration Between Ontologies having Structural Differences. International Journal on Artificial Intelligence Tools 20(6), 1127–1156 (2011)
6. Henderson-Sellers, B., Gonzalez-Perez, C.: Standardizing Methodology Metamodelling and Notation: An ISO Exemplar. In: Kaschek, R., Kop, C., Steinberger, C., Fliedl, G. (eds.) UNISCON 2008. LNBIP, vol. 5, pp. 1–12. Springer, Heidelberg (2008)
7. Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., De Bo, J., Dieng, R., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Maynard, D., Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaiko, P., Tessaris, S., Van Acker, S., Zaihrayeu, I.: State of the art on ontology alignment. KnowledgeWeb project deliverable D2.2.3, v.1.2 (August 2, 2004), `http://knowledgeweb.semanticweb.org/`
8. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to match ontologies on the Semantic Web. International Journal on Very Large Data Bases 12(4), 303–319 (2003)
9. Tversky, A.: Features of Similarity. Psychological Review 84(4), 327–352 (1977)
10. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The Alignment API 4.0. Semantic Web 2(1), 3–10 (2011)
11. Davidovsky, M., Ermolayev, V., Matzke, W.-E., Tolok, V.: Evaluation of Semi-automated Ontology Instance Migration. In: Essaaidi, M., Malgeri, M., Badica, C., et al. (eds.) Intelligent Distributed Computing IV. SCI, vol. 315, pp. 179–190. Springer, Heidelberg (2010)

Cognitive Modeling and Support for Ambient Assistance

Judith Michael, Andreas Grießer, Tina Strobl, and Heinrich C. Mayr

Alpen-Adria-Universität Klagenfurt, Universitätsstraße 65-67, 9020 Klagenfurt, Austria
{Judith.Michael,Andreas.Griesser,Tina.Strobl,
Heinrich.Mayr}@aau.at

Abstract. The aim of the Human Behavior Monitoring and Support (HBMS)[1] project is to learn about the individual skills and behavioral knowledge of a person in order to support that person when needed. It is intended as a contribution to enable elderly people to live autonomously in their domestic environment as long as possible. The basic idea is to build a cognitive model of the behavior of a person while she/he is of sound mind and memory. In case of mental incapacitation this model will be used as a knowledge base for generating support information. The paper outlines the first results of the HBMS project with a focus on the investigative survey and the overall architecture of the chosen approach.

Keywords: Cognitive Modeling, Ambient Assistance, Model Integration.

1 Introduction

We are facing unprecedented demographic changes in recent years. The European population aged 60 and above is rising by more than 2 million per year [1]. Life expectancy is increasing sharply, and so is the number of people who will need care—which will lead to exploding healthcare costs [2]. The working population is decreasing, and so we have to find solutions to the problem of how care for the elderly will be handled without enough human resources available.

Ambient Assisted Living (AAL) is a research area that seeks to develop methods to support elderly people in their everyday life. Steg's definition is instructive [3]:

> *"AAL aims to prolongate the time people can live in a decent way in their own home by increasing their autonomy and self-confidence, the discharge of monotonously everyday activities, to monitor and care for the elderly or ill person, to enhance the security and to save resources." (p. 28)*

The main aim of AAL is to enable the elderly to live longer and as autonomously as possible in their domestic environment, thus decreasing healthcare costs and helping them to reach higher satisfaction with their quality of life.

Previous research has concentrated on usability and security aspects of technical devices, but so far, little attention has been paid to cognitive support. Our project,

[1] Funded by the Klaus Tschira Stiftung, Heidelberg.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 96–107, 2013.

Human Behavior Monitoring and Support (HBMS) [4], whose goals we present in Section 2, aims at filling this gap. In Section 3 we discuss a psychological survey case study that we used to explore the needs of the elderly. The main question was to determine aspects of everyday life where people might require support and areas where they do *not* want help. We present preliminary results based on a specific questionnaire and interviews with a number of elderly people. In Section 4 we present a prototype system as a proof-of-concept. We conclude by outlining how our findings could be used in the future and defining HBMS as a contribution to support elderly people who want to live longer and more autonomously in their own environment.

2 HBMS: Human Behavior Monitoring and Support

The main idea of the Human Behavior Monitoring and Support (HBMS) project is to support people in their everyday life [5] when cognitive functions like, e.g., memory are decreasing. This might happen when we are getting older and the brain is suffering from neurobiological changes. Such decrease can also happen independent of age, for example if we are stressed or unpracticed in doing specific activities.

HBMS wants to support people by retrieving forgotten knowledge. Therefore it is necessary to build a cognitive model of a person's behavior while she/he is of sound mind and memory. In case of mental incapacitation this model will be used as a knowledge base for generating support information. HBMS will not compare the behavior of different people as some studies do (see [6]) and it will not suggest how someone should act in a particular situation compared to others. The main objective is to retain information about how a person performed actions herself or himself before, not how a third person performs an action (which in most cases could be different).

2.1 Aim and Target Groups

The aim of the HBMS project is to learn about individual skills and a person's behavioral knowledge in order to support that person later when needed.

Individual behavior related to observed scenarios is mapped to sequences of actions. If sequences of the behavior are missing in the daily routine, it will be possible to support the person with his/her own knowledge and in his/her own words.

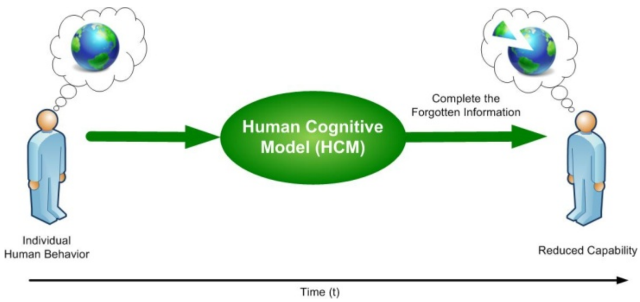


Fig. 1. Support by a person's own (former) knowledge

Fig. 1 shows that the Human Cognitive Model (HCM) tries to complete missing steps of activity or other information for a person. The missing information, e.g., which button is the next one to descale the coffee machine, or how to program the video recorder, could be offered if necessary.

The target group for the project is in fact everybody who could benefit in the future from being reminded of their former, forgotten knowledge, independent of the person's age. As already indicated above, we often forget things if we are stressed, overtired, or untrained when we perform actions outside of our ordinary routine. Consequently there is practically no restriction to possible user groups for HBMS. In detail the target groups have different requirements regarding their age, their circumstances in life, or individual needs. Therefore we decided to choose a more general approach. The individualization is reached by using HBMS and learning the behavior and ontology of each individual person.

We focus on scenarios from four different areas of everyday life: the use of technical devices, activities of daily life, business processes, and ambient environments.

2.2 The Process

Fig. 2 illustrates the HBMS process for an individual person. It is possible to preserve the individual memory of a person by building a cognitive model of her/his behavior.

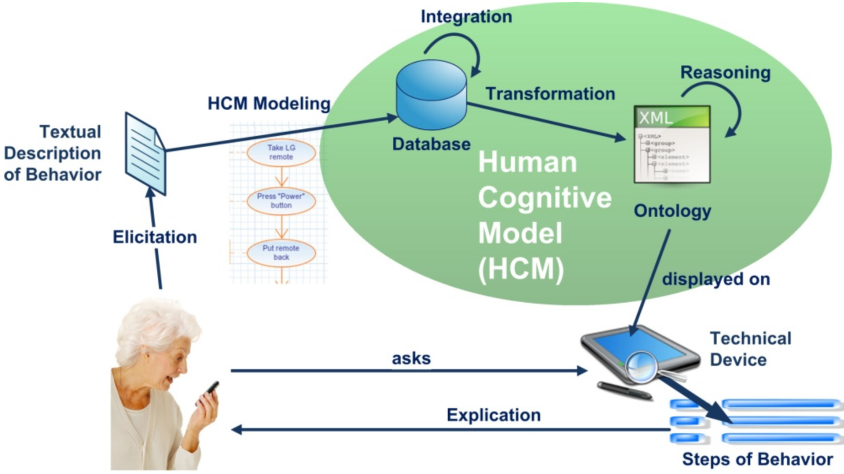


Fig. 2. The HBMS process

Observation will be done automatically in a later project phase, e.g., activity and intention recognition [7], the usage of smart meters to detect and individualize Activities of Daily Living [8] or the usage of sensors and Case Based Reasoning (CBR) for activity recognition [9]. At the first stage a psychology domain expert observes a person and writes down the different steps of action, e.g., as they use a coffee machine. Behavioral sequences of the person's Universe of Discourse are elicited step-by-step, mapped to and integrated into a cognitive model (HCM, Human Cognitive Model). The meta-model of the HCM is created based on concepts of the Klagenfurt Conceptual Predesign Model (KCPM) [11], which are adapted to the structure of human behavior and human intention [12]. In the long term we plan to refine it with techniques from CBR [13]. For model integration we will reuse and adapt methods from [10] to the new requirements of HBMS.

This model then is transformed to a formal ontology which allows inferring, by reasoning, new knowledge as the given situation may demand. If an individual needs support for an action, HBMS can return information about missing steps out of his/her former knowledge, e.g., on a technical device like a smartphone, tablet PC, or perhaps in the future some other kind of "smart environment."

3 Empirical Study

A central question that needs to be addressed in the context of AAL is what people with reduced cognitive capability need to cope in their everyday lives. We gained this information by including future users. We used a number of methods, including a questionnaire, a workshop with students of all ages and analysis of results from other scientific studies of this topic. We gathered preliminary data from December 2011 to March 2012. We sent the questionnaire to all students of the Alpen-Adria-Universität Klagenfurt. During the study period, 203 persons of all ages participated. Fig. 3 shows the detailed age distribution.

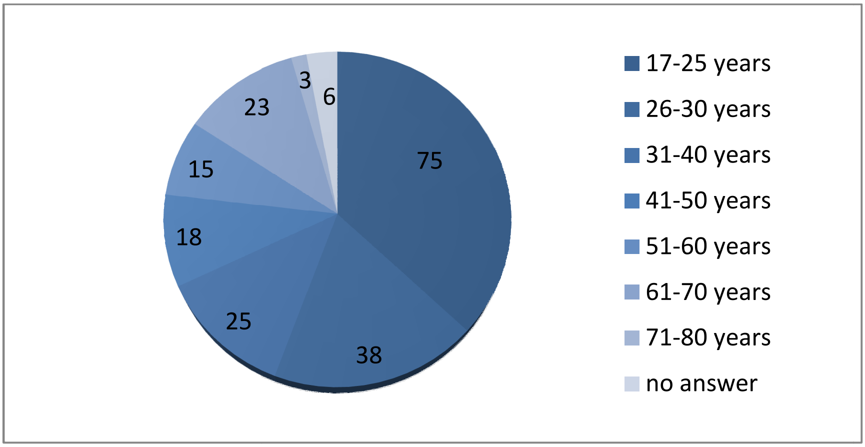


Fig. 3. Age distribution of the survey participants [values in absolute count]

Predictably, the age class under thirty years is overrepresented (55%) which correlates to the development of the student structure. The results of all ages are interesting to find out if there are differences between age classes and their attitudes toward technology.

Our results so far seem to suggest that there is interest in a system that offers cognitive support on multifunctional devices, especially for businesses like online government, and reminders on contents of conversations, dates, or the shopping list. Another conclusion at this point is that activities that people do not repeat regularly are most vulnerable to being forgotten.

3.1 The Survey

We performed an investigative survey to find out which kinds of support might be useful for users and how it should be provided.

One result from the questionnaire is that people believe the use of e-banking and e-government processes is becoming increasingly important. However, these processes are often difficult to handle and hard for customers and citizens to remember. At the same time, the number of office workers available to help will decrease.

Not only are technical devices getting increasingly complex, our everyday life does as well. Schedules, dates with other persons, deadlines for submitting applications, taking medicine at a specific time, and reminders for the shopping list are matters people would like to be reminded of when dealing with them. These results correlate with those received in Berlin through a voluminous and highly qualified study [14]: memory, knowledge and perception speed are skills that are often progressively diminishing and could adversely affect the ability to cope with everyday life. While our conclusions remain tentative, as another result the data reveal a basically positive attitude to technology for all age classes in our study.

Nevertheless, these results must be viewed with a critical eye: the students were invited and not forced to participate the survey, and it is likely that those students who are not interested in technical processes at all did not take part. However, what we can say is that there is no difference between age and attitude to technology for those people who participated.

To show some possible scenarios for HBMS systems, we defined two scenarios indicating how the system could be used in the future. Maria, a fictional 75-year old person has problems using her video recorder. The question given to the participants was if they could imagine that an acoustic signal, e.g., a voice, is guiding Maria through the right order in which to press the buttons. As shown in Fig. 4, 87.25% of the participants were able to imagine that scenario and would like to have this possibility for themselves too.

Comments given to this scenario were that it would be useful because of the easy and fast support, but there was also a desire to switch the system off if it is not needed and to change the data output, e.g., not a voice but support on the screen of the television.

At the same time, 6.37% of the participants do have problems with this scenario and could not imagine being supported by an automated system in this case. The most common criticism was that family contact could be reduced as a consequence of the decreased need for help.

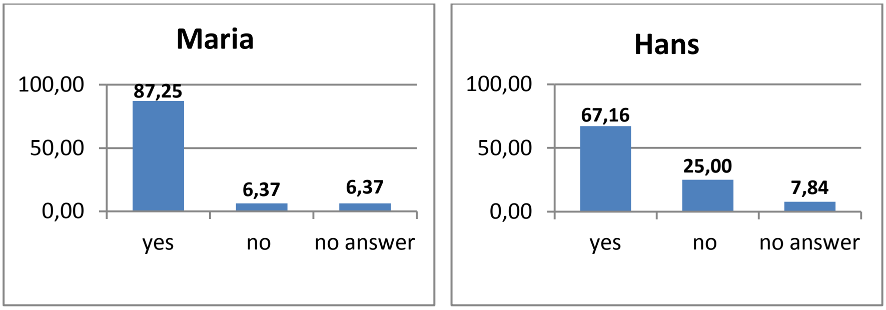


Fig. 4. Example of Maria and Hans

The second scenario was about Hans, a fictional 50-year old man whose local bank branch was closed and so he had to do his banking business via Internet. If he had problems with it, he could be supported by a voice, e.g., telling him about the next steps necessary to finish the transactions. 67.16% of the participants could imagine this scenario for themselves too, as seen in Fig. 4.

Most people who refused this scenario criticized possible security problems w.r.t. these sensitive data. Interestingly enough, two thirds of participants did not worry about this problem.

Performing online financial transactions is becoming increasingly important. 63% of the participating students are already using e-banking for their everyday needs. In any case, almost every fourth participant answered that he/she is reliant on help when using online banking.

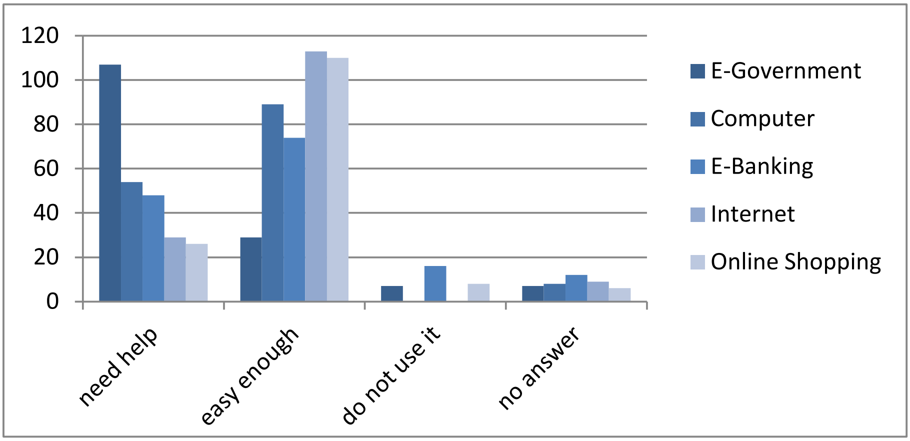


Fig. 5. Possible areas of support

Fig. 5 shows for which areas the participants find help necessary and which are easy enough to use: using the Internet, doing online shopping or using the computer seems quite easy. The use of e-government services turned out to be most difficult (registration inquiries, registration of dogs, communal tax declaration).

Using domestic technical equipment seems to be easier. Fig. 6 shows that study participants believed turning on the television or the washing machine to be simple. Every fourth participant indicated the need for help using a mobile phone, and we confirmed this in the interview phase of our study as well. The most frequent answers were that (1) smartphones have too much functionality so that it is difficult to execute the desired function, and (2) that infrequent actions (like changing the password or ringtone) are forgotten quickly.

A similar example given by a 62-year old workshop participant was her experience with using her digital camera. Using it at the standard settings is no problem, but changing the white balance is not simple for her.

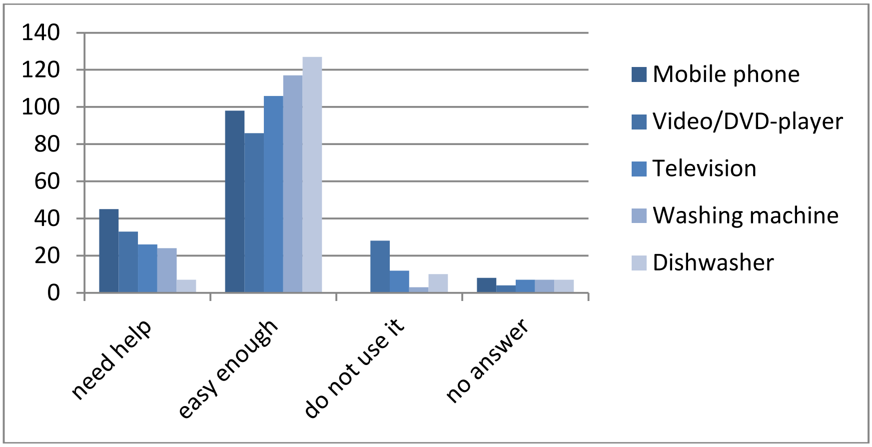


Fig. 6. Areas of support for domestic technical devices

3.2 Conclusion

As we have seen, there is considerable interest in a support system as intended by our project, HBMS. However, potential users expect that it should be easy to switch off such a system when they do not need or want it. Ways in which we should present output data of such a system to users is not yet completely clear: we will learn more about that when testing and using the prototype.

At the current stage of the project we have focused on support for using a coffee machine, monitoring the usage of a washing machine, and providing appropriate assistance for e-government services as early test scenarios. In a next step we will add further scenarios mentioned in the questionnaire and we will incorporate ideas that came from workshop participants.

Based on the response to the survey we decided to add information to the Human Cognitive Meta-Model (HCMM). Users want data like pictures or videos included in behavioral steps. Voice output, as proposed in the scenarios for Maria and Hans, is vital. Users should also be able to choose between acoustic, visual, or other support in order to meet individual preferences and needs. Moreover, participants strongly demand the ability to customize such a system. We need to provide this, which means including customization elements in the data schema.

4 The Prototype System

The questionnaire and workshop gave us deeper insight into the people's needs. But a questionnaire alone is not enough to evaluate our ideas. Even though implementation is not the focus of our research, we developed a prototype system as a proof-of-concept. With the prototype it is possible to show potential future users how this system might help them. The HMBS system contains the modeler tool, integration and transformation applications, and explication services in different domains.

4.1 The Architecture

The HBMS system is based on a client-server architecture (see Fig.7). At the client side we distinguish the modeling, network, and mobile domains. On the server side are the integration, ontology, and service domains.

The standalone HCM Modeling Tool implementation is based on the Eclipse Rich Client Platform (RCP) [15]. A set of very useful editing features, such as automatic layout and aligning, drag-and-drop, and direct editing support are provided by the Eclipse-based graphical framework Graphiti. The representation concepts of the Human Cognitive Modeling Language (HCML) are fully supported by the modeler. The Hibernate framework is used as persistence layer for retrieving and storing the HCM model in a relational database. The integration of additional observed behavioral sequences to the generalized HBMS model using Case-Based Reasoning (CBR) and Natural Language Processing (NLP) is performed automatically.

In the ontology domain, data is retrieved from the relational database through Hibernate and then transformed into the Web Ontology Language (OWL) [16]. The transformation is done by server-side processing using transformation techniques like QVT (Query/View/Transformation) or ATL (ATLAS Transformation Language) [17]. The resulting OWL files generated by the JENA Framework [18] are stored in the file system on the server. The ontology editor Protégé is used for prototype development and visualization of these ontologies.

In the service domain, the data to support the individual user are processed. As in adaptive demotic systems the OSGi framework is used to allow uniform access to services [19]. The implemented OSGi bundles are organized in a layered architecture. The base tier is the persistence layer where the OWL files are stored in the Knowledge Base Bundle.

The middle tier represents the processing logic where the result of the query is converted into a generic step-based data structure. The service tier is the presentation layer which provides public services that can be accessed by appropriate devices. The mobile domain includes all mobile devices, e.g., smartphone, tablet PC, and notebook with wireless network. The network domain characterizes all devices (workstations, terminals) in the local network.

To ensure comfortable and versatile use, mobile devices are currently used as the output medium. Communication with server components runs via web interfaces to allow an appropriate presentation on the end-user devices.

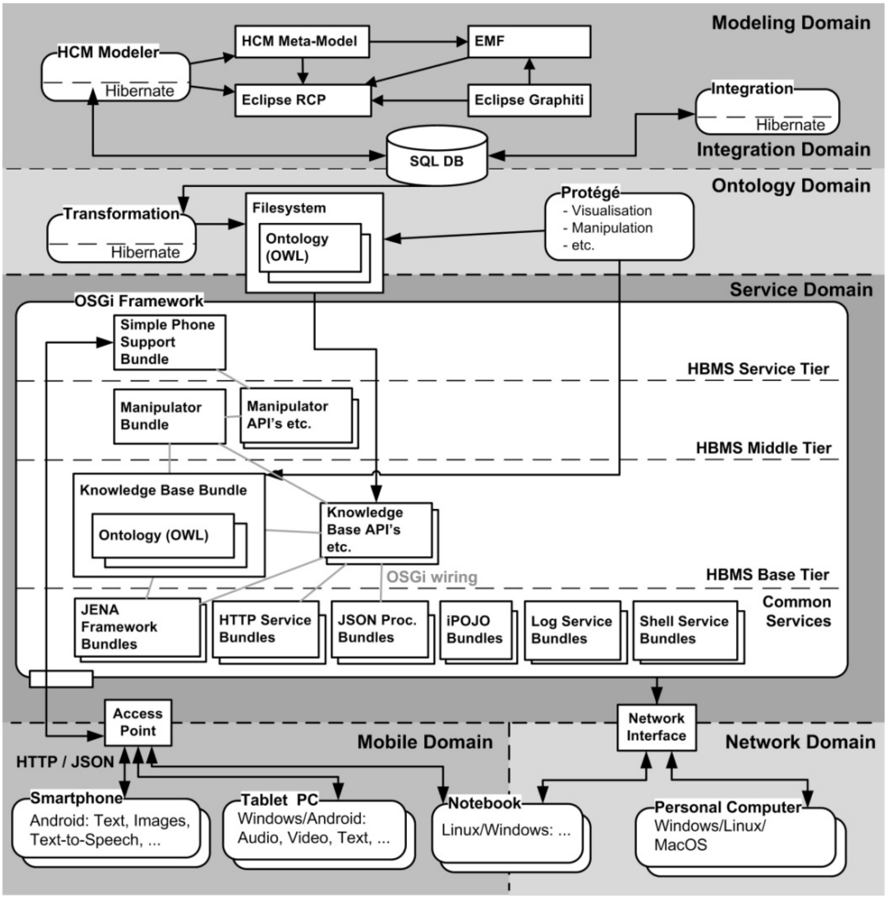


Fig. 7. Architecture of the HBMS Prototype

4.2 Tools and Applications

Related to the four scenarios mentioned in Section 2.1, we map a person's individual behavior steps to sequences. This model contains static and behavioral aspects. To do this, we use our know-how in conceptual modeling (KCPM, Klagenfurt Conceptual Predesign Model [11]) and evolve the KCPM concepts to cognitive modeling. The Human Cognitive Modeling Language (HCML) is still under development. More information can be found in [20].

Fig. 8 shows a screenshot of the HCM Modeling Tool prototype. In this application it is possible to model behavioral sequences as well as static components and to make them persistent in the database.

As with [10] and [21], we integrate the sequences in a background process to a generalized model. Starting from this point, the integrated, generalized, and enriched model is a Human Cognitive Model (HCM). The transformation from HCM, an ontology language for behavioral and episode description, into a suitable representation language is also a background process. Due to increasing popularity of the Semantic Web, we chose OWL as the ontology representation language.

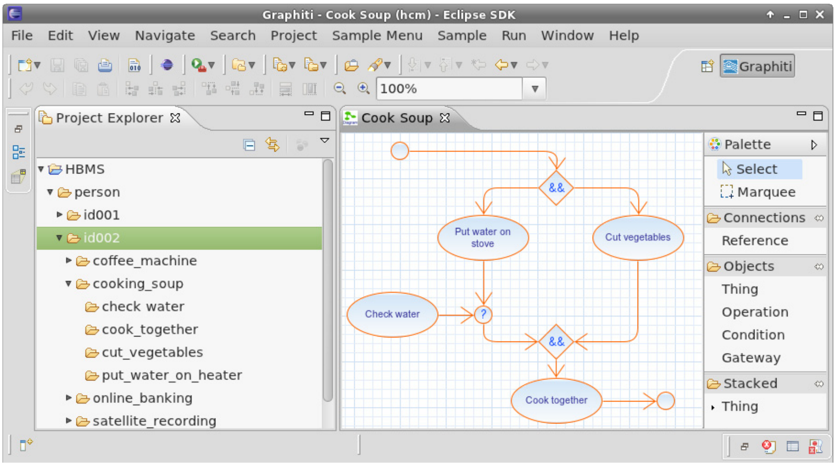


Fig. 8. Screenshot of the HCM Modeling Tool

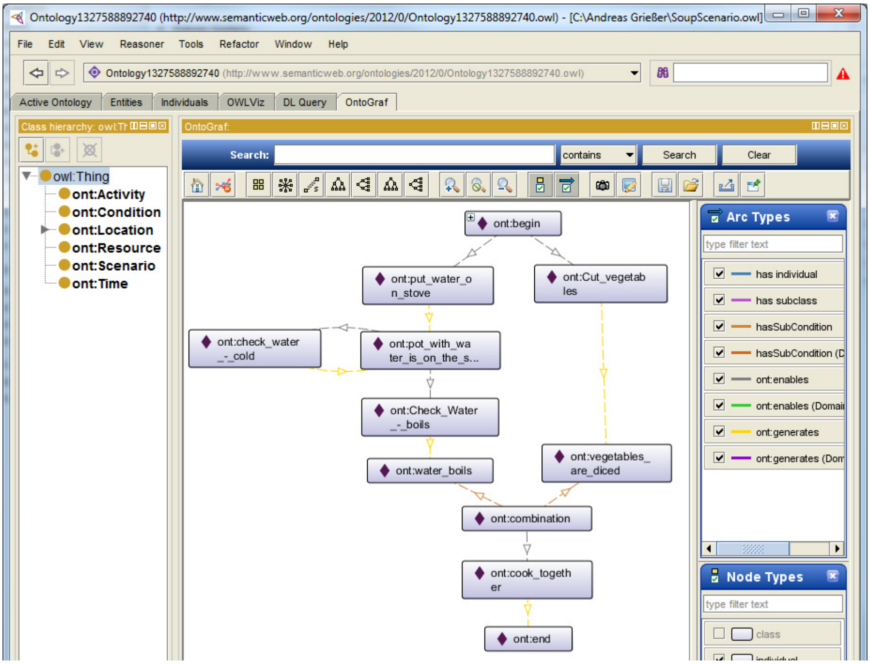


Fig. 9. Soup-cooking ontology visualized in Protégé

Ontologies as explicit formal specifications of conceptualizations include integrity and inference rules that need to be derived. OWL is desirable for this purpose as it has the ability to express semantics, constraints, and individuals of the problem domain. The transformation of UML activity diagrams to OWL has been examined in [22] and can be successfully applied to our project in a similar way. For lossless and consistent transformation of HCM, we will also develop behavioral model mapping rules.

Fig. 9 shows the ontology that results from transforming the HCM model given in Fig. 8. Reasoning allows us to derive logical conclusions and new knowledge from known behavioral steps. Several implementations of so-called reasoners are freely available in third-party libraries [23].

5 Summary and Outlook

Challenges for our society lead to interesting and exciting topics for research like Ambient Assisted Living. To sum up, with HBMS we propose a solution for everybody's need for independence from third parties while performing certain activities. We are working on methods for creating a cognitive model and using it as a basis for situational support. Our prototype enables us to elicit and model behavioral steps and to integrate them into a Human Cognitive Model which then is transformed to a formal ontology. As a last step, information requested about missing steps can be returned for an individual person, e.g., on a mobile device, tablet PC or in a smart environment. Our prototype will be presented to several groups of end users. Their feedback will be used for further development within the project.

Ongoing and further work includes the exploitation of methods from Natural Language Processing for modeling [21] and Case-Based Reasoning [9]. There are many alternatives for rendering generated support information, e.g., via mobile devices, especially w.r.t. emerging opportunities for speech analysis and synthesis, but also via specific in-room multi-media systems. Multimodal interaction will be provided to make the use of HBMS accessible and intuitive.

Acknowledgements. The authors would like to thank the reviewers for their helpful comments and Prof. Steve Liddle for his help to improve our English.

References

1. Publications Office of the European Union: Demography Report 2011: Older, more numerous and diverse Europeans. Commission Staff Working Document (2011), `http://ec.europa.eu/social/main.jsp?catId=502&langId=en`
2. Berting-Hüneke, C.: Selbständigkeit im Alter erhalten. Springer, Berlin (2002)
3. Steg, H., Strese, H., Loroff, C., Hull, J., Schmidt, S.: Europe Is Facing a Demographic Challenge. In: Ambient Assisted Living Offers Solutions. VDI/VDE/IT, Berlin (2006)
4. Griesser, A., Michael, J., Mayr, H.C.: Verhaltensmodellierung und automatisierte Unterstützung im AAL Projekt HBMS, 5. Deutscher AAL-Kongress, Berlin (2012)

5. Bucks, R.S., Ashworth, D.L., Wilcock, G.K., Siegfried, K.: Assessment of Activities of Daily Living in Dementia: Development of the Bristol Activities of Daily Living Scale. Age and Ageing 25, 113–120 (1996)
6. Hoffmeyer, A., Yordanova, K., Teipel, S., Kirste, T.: Sensor based monitoring for people with dementia: Searching for movement markers in Alzheimer's disease for a early diagnostic. In: Workshop on User Interaction Methods for Elderly People With Dementia at the AmI, vol. 11 (2011)
7. Bader, S., Kirste, T.: A Tutorial Introduction to Automated Activity and Intention Recognition. In: Lecture Notes for the Interdisciplinary Colleg, IK (2011)
8. Clement, J., Ploennigs, J., Kabitzsch, K.: Smart Meter: Detect and Individualize ADLs. 5. Deutscher AAL-Kongress, Berlin (2012)
9. Zhou, F., et al.: A Case-Driven Ambient Intelligence System for Elderly in-Home Assistance Applications. Institute of Electrical and Electronics Engineers, New-York (2011)
10. Vöhringer, J., Mayr, H.C.: Integration of schemas on the pre-design level using the KCPM-approach. In: Nilsson, A.G., Gustas, R., Wojtkowski, W.G., Wojtkowski, W., Wrycza, S., Zupancic, J. (eds.) Advances in Information Systems Development: Bridging the Gap between Academia & Industry. Springer, Heidelberg (2006)
11. Mayr, H.C., Kop, C.: A User Centered Approach to Requirements Modeling. In: Glinz, M., Müller-Luschnat, G. (eds.) Modellierung 2002, Köllen, Bonn. Lecture Notes in Informatics (LNI), vol. P-12, pp. 75–86 (2002)
12. Soffer, P., Rolland, C.: Combining Intention Oriented and State based Process Modelling, Conference on The Entity-Relationship Approach (ER), Klagenfurt, Austria (2005)
13. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7, 39–59 (1994)
14. Reischies, M., Lindenberger, U.: Grenzen und Potentiale kognitiver Leistungsfähigkeit im Alter. In: Die Berliner Altersstudie. Akad.-Verl., Berlin (2010)
15. McAffer, J., Lemieux, J.: Eclipse Rich Client Platform: Designing, Coding, and Packaging Java(TM) Applications. Addison-Wesley Professional (2005)
16. Antoniou, G., Van Harmelen, F.: Web Ontology Language: OWL. Handbook on Ontologies (2004)
17. Jouault, F., Kurtev, I.: On the Architectural Alignment of ATL and QVT. In: Proceedings of ACM Symposium on Applied Computing (SAC 2006), Model Transformation Track, Dijon, Bourgogne, France (2006)
18. Apache Jena™, `http://jena.sourceforge.net/`
19. Wu, C., Liao, C.: Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 193–205 (2007)
20. Michael, J., Bolshutkin, V., Leitner, S., Mayr, H.C.: Behavior Modeling for Ambient Assistance. In: Proceedings of the International Conference on Management and Service Science, MASS 2012 (2012) (in press)
21. Bellström, P., Vöhringer, J.: Towards the Automation of Modeling Language Independent Schema Integration. In: Proceedings of the 2009 International Conference on Information, Process, and Knowledge Management, pp. 110–115. IEEE Computer Society Press (2009)
22. Khan, A., Minhas, A., Niazi, M.: Representation of uml activity models as ontology. In: Proc. of 5th Int. Conference on Innovations in Information Technology (2008)
23. Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoned Web Semantics: Science. Services and Agents on the World Wide Web 5(2), 51–53 (2007)

Architecture for Social Life Network to Empower People at the Middle of the Pyramid

Athula Ginige[1], Tamara Ginige[2], and Deborah Richards[3]

[1] School of Computing, Engineering & Mathematics, University of Western Sydney, Australia
a.ginige@uws.edu.au
[2] Department of Computing, Faculty of Science, Macquarie University, Australia and
Faculty of Business, Australian Catholic University, Australia
tamara.ginige@acu.edu.au
[3] Department of Computing, Faculty of Science, Macquarie University, Australia
deborah.richards@mq.edu.au

Abstract. We have designed an architecture for a mobile based information system to empower people in the middle of the pyramid (MOP). MOPs is now half of the world's population has only a mobile phone to get connected to the Internet. They need applications to enhance their livelihood activities. These applications need to provide information on rapidly changing dynamic situations such as fluctuations in market prices, prevailing supply and demand situation for their produce as well as more stable information such as information on seeds, pests, weather patterns, soil types etc. Our architecture is designed to provide dynamic information by aggregating micro-blogs, status updates and data from sensors. It also has connections to established data sources and websites to provide more stable information and tools for empowerment.

Keywords: Social Life Networks, Mobile based Information Systems, micro-information aggregation, User empowerment.

1 Introduction

Currently social networks, designed with the primary goal of facilitating communications among people using emerging technologies, have been growing in popularity in the last few years. These networks provide us with tools to share our experiences using text and a variety of other mediums. The ways users can participate in these networks have significantly changed as a result of the second generation of Web applications known as Web 2.0. These applications allow users not only to view content but also to add content. Today users can use micro-blogging to communicate and can attach photos and videos to it. Users who were previously passive information consumers have become the information producers which can be seen as an empowerment of previously passive information consumers [1].

The micro-blog concept has made it possible to use mobile phones to access social networks. At present, 90% of the world population is covered by a mobile signal, 76%

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 108–119, 2013.

of the world population has a mobile subscription and in developing countries the subscription rate is 67% [2]. Today mobile phones are equipped with a touch keyboard, range of sensors such as cameras, GPS, and have the ability to run applications like music players, photo albums etc. These Smartphone sales to end users reached 54.3 million units, an increase of 48.7% from the first quarter of 2009 according to Gartner, Inc. [3].

The success of current social networks and the explosive growth of mobile applications and mobile devices with myriads of sensors have enabled users to be socially connected while they are mobile. Jain [4] has shown that next generation of social networks can be designed not only to connect people to share information and experiences, but also to connect people with others and essential information resources by making use of this array of sensors. He refers to these new networks as Social Life Networks (SLN).

Jain has used a technology driven perspective to show that the world population can be classified into three distinct classes in a pyramid. People at the top layer of the pyramid (about 1.5 billion) have access to modern telecommunication networks using their computers and Smartphones; at the bottom of the pyramid (about 2 billion) are the people deprived of any modern means of communication. At the Middle of the Pyramid are about 3.5 billion people who have mobile phones but are not part of the modern Internet due to lack of useful applications. This group of people are referred to as MOPs. The MOP has been growing and is now about half of the world's population. Most of these people live in developing countries.

At present most mobile applications have been developed with a focus on the needs of the top 1.5 billion of the world population. While MOPs are ready in terms of technology and connectivity, they do not have useful applications specifically designed for their needs at present; developing mobile applications for this segment of the market opens up a new market that is twice the size of the current market.

In his theory of motivation, Abraham Maslow [5] proposed a hierarchy of needs based on two groupings: deficiency needs and growth needs. Deficiency needs are physiological (hunger, thirst, bodily comforts, etc.), safety/security (out of danger), belongingness and love (affiliate with others, be accepted) and esteem (to achieve, be competent, gain approval and recognition). Within the deficiency needs, each lower need must be met before moving to the next higher level. According to Maslow, an individual is ready to act upon the growth needs if and only if the deficiency needs are met. Current Social Networks enable users to belong to groups, being loved and recognised by the others. For MOPs we need a new kind of networks that address physiological and security related needs.

By connecting people to resources using information aggregated from sensors, SLNs can be made to become very useful mobile based information system (MBIS) to support livelihood of MOPs and it may help to address some of the issues related to satisfying their deficiency needs.

In this paper we take Jain's ideas further and present a high-level logical architecture for a SLN to provide timely information to support livelihood activities of MOPs as well as tools for them to actively participate in the information creation and consumption process. Section 2 describes the research process that we used to develop this architecture. The third section describes the analysis phase to get a deeper understanding of related work and user's information needs to support their livelihood activities. Section 4 presents potential mobile applications based on the findings from the analysis phase. A conceptual architecture is presented in section 5. In section 6 we explore ways to empower users and in section 7 we present a high-level logical architecture derived from the conceptual architecture that also includes tools to empower users. The last section contains the current status of the project, future work and the conclusions.

2 Research Approach

In order to develop the architecture, first we had to discover the essential requirements for a mobile based information system that will optimally benefit MOPs. It was not possible to identify these essential requirements by surveying a representative sample of users as they have not experienced the potential of such applications. Thus we used an approach consisting of the following phases to arrive at a high-level logical architecture for a Social Life Network specifically aimed at empowering MOPs.

Analysis Phase: In this phase we reviewed the mobile applications that have been developed by others to assist day to day activities of people in developing countries to get an understanding of the functionality, success and limitations of these applications, met with focus groups to understand their information needs and reviewed technologies that can be used to meet the identified information needs.

Identify Potential Applications: In this phase, based on findings from the analysis phase, we identified a couple of ways in which MBIS can be effectively used to enhance livelihoods of MOPs.

Conceptual architecture design phase: In this phase we designed a high-level conceptual architecture for the identified applications.

Study of user empowerment: In this phase we studied the theories that support empowerment. With the understanding of the information needs of the focus groups, we identified some processes that can empower the users.

Logical architecture design phase: The high-level conceptual architecture was then refined to include the required characteristics for user empowerment and a high-level logical architecture for the system was designed.

3 Initial Analysis

In the initial analysis phase we performed three major activities.

3.1 Review of Related Mobile Applications

In many developing countries agriculture and fisheries play a major role in the country's economy. We found examples of mobile phone usage to achieve a better market dispersion.

In his study, Jensen [6] investigated the effect of mobile phones on the fishing industry which is one of the most important industries in Kerala, India. Fishermen mostly sell their catch locally as they do not have money to invest in proper storage and also due to difficulties in taking them to far away market places due to high transportation costs. More importantly, local fishermen do not have any information on their market conditions such as the competition, other suppliers and buyers. As a result, there was high inefficiency in the process, price dispersion and wastage. After the introduction of mobile phones, Jensen observed that the fisherman were able to exchange information about the market better and as a result, price dispersion was reduced, fish were allocated efficiently across the market and waste was reduced. He has shown that this leads to important welfare improvements for both fishermen and consumers; fishermen's profits increased by 8%, consumer prices decreased by 4% and consumer surplus increased by 6%. Aker carried out a similar study to investigate the effect of mobile phone coverage between 2001 – 2006 on the grain market [7]. She found that the introduction of mobile phones reduced price dispersion by 10%. As a result, consumers benefitted with lower prices and suppliers gained higher profits.

There are many mobile applications that have been developed to deliver information to users. Tata Consultancy Services Ltd in India offers a service called mKrish that delivers crop advice to farmers in rural India by cell phone [8]. China mobile sends farmers information about planting techniques, pest management and market prices via their website 12582.com. TradeNet in Ghana links buyers and sellers of agricultural products in nine African countries and CellBazaar provides a text-based classified ad service in Bangladesh [9].

Mobile communications have been used to improve the health systems in many developing countries. In Uganda and Mozambique, new mobile diagnostic equipment have been developed to address the issues related to lack of health workers, equipment and tools [10].

There are a number of studies that have been done to implement mobile educational applications in developing countries. Muyinda [11] has created a framework for developing mLearning applications for learners in Africa. Both Kumar [12] and Power [13] have carried out studies to investigate the opportunities that mobile devices provide in learning and teaching in India and Bangladesh respectively. Even though there are still issues such as infrastructure, affordability and availability of mobile technologies in developing countries, these authors indicate that mobile devices are ideal vehicles for making educational opportunities accessible to rural children in developing countries.

Mobile banking is another application area where many applications have been developed in recent years. M-PESA, a mobile phone-based money transfer service, boasts more than nine million users in Kenya [14]. This mobile application was officially introduced onto the Kenyan market in 2007 by Safaricom, the Kenyan mobile service provider. It facilitates numerous financial services such as checking account balances, making deposits and withdrawals, transferring money and phone credit to other users.

This review provided insights into information needs of the MOPs. It also revealed that most of the applications are designed to provide information to MOPs, but there were no facility for MOPs to provide information that can be shared with a community and create a collective knowledgebase which is essential for empowerment.

3.2 Focus Group Meetings to Find Out Information Needs of Users

We conducted a focus group meeting with a group of people in a Sri Lankan rural village. In Sri Lankan economy, agriculture is the most important sector and approximately 33% of the total labour force is engaged in agriculture [15]. Farming rice, vegetables and other crops is the most important activity for the majority of people living in rural areas of Sri Lanka. Currently there are not many mobile applications that have been developed to address the local needs of the farmers.

The focus group consisted mainly of farmers and people who have small art and craft businesses. The farming system offers self-sufficiency to rural families in Sri Lanka. The definition of a farm can vary depending on the size of the land that they have. It could be a large to medium sized land that is situated away from their homes or it could be their small backyard. No matter what the size of the land is, it is very common for these people to grow vegetables, fruits, flowers and spices in their farms. They will sell these items to bring an income to their families.

A major observation during this focus group discussion was the difficulties that these people faced in finding the current market conditions for their products. Therefore, they were not certain about the best place to take their produce, the best time to pick them and the best price that they can get to gain a profit. If they had known that the market prices were not very competitive on a certain day, then they could have waited a couple of days before picking their produce. Farmers who have a small quantity of produce have experienced difficulty to find appropriate space in the market to sell their produce as the farmers who produce large quantities dominate the market space. In some cases, farmers sell their produce to the middle person and often the middle person takes the advantage from these situations as farmers do not have enough information to find a buyer and carry out successful negotiations. As a result, farmers do not gain much profit from their produce and do not see that their livelihood is improving. Another area where farmers need help is advice on weather conditions, fertilizer, diseases and how to prevent them.

From the focus group findings we identified MOPs will need two types of information to support their day-to-day activities. One type is the information related to their field of work. For example, farmers wanted to know about different types of fertilizer, common diseases and how to address these, and how to carry out farming related activities. This type of information is relatively stable (static) and often can be found in books, websites, etc.

The second type of information relates to current market prices, prevailing supply and demand status for a product, etc. This information is constantly changing, hence dynamic. Therefore for a mobile based information system to meet the user information needs it should be capable of providing both of these types of information to the users.

3.3 Investigation of Available Technologies to Meet User Information Needs

In the previous section we identified the need for two types of information. The system should have access to relevant databases or websites to obtain the static information that the users will need to support their day-to-day activities. Sometimes to respond to a user query, it would become necessary to synthesise information obtained from multiple data sources.

The second type of information the users will need is on the current status of a range of things such as current market prices, current demand for a particular product, current supply situation, etc. Jain [4] has identified a variety of sensors that are available to capture useful multimedia data about human life. He categorised these sensors into two main classes, i.e. physical sensors and human sensors. Cameras, microphones, GPS and motion sensors are examples of physical sensors. The sensory data generated from these physical sensors can be in many different formats, e.g. image, video and audio. The raw multimedia data usually needs to be processed to produce enough knowledge for human beings to understand.

On the other hand humans, with the intrinsic intelligent sensory system inside their body, are naturally good at sensing the world around them and creating knowledge out of it. The output from the human sensing and knowledge creation process can be documented in various representative formats, e.g. writings, paintings, music, and movies.

Jain and his research team have developed a way to combine heterogeneous data coming from all types of human and physical sensors into a common spatio-temporal-thematic representation [16]. Their system is shown in figure 1. The information from sensors is first sent to the "Aggregation and Composition" module. In this module a "theme" is assigned to the information and mapped on to a 3-dimensional array based on time, latitude and longitude coordinates. In the "Situation Detection" module various two dimensional data processing algorithms such as summation over a time interval and correlation algorithms are used to detect evolving situations. Based on pre-set conditions alerts are generated when a corresponding event is detected. Also there is a query interface to get high-level context extracted from aggregating micro information. For example if micro information that fits the theme "have flu" is aggregated over a period of time during a flu epidemic, one can obtain a picture in real time how flu is spreading in a country.

Fig. 1. High-level view of the micro information aggregation framework. SOURCE:[4]

4 Potential Applications

Based on the findings from the analysis phase, we have selected two problem areas to investigate how a MBIS can assist in providing a solution to enhance the livelihood activities of MOPs. We have chosen one area from agriculture and the other from the fishing industry as they both play major roles in the economies of many developing countries.

4.1 Assisting Agriculture via Informed Selection of Crops for Cultivation

Due to lack of access to current information, often farmers tend to grow the same crop within a region, and this could cause potential over supply of crops [17]. But farmers come to know, or realise, there is an oversupply only when they bring their harvest to the market, and the oversupply reduces market price for the crop, disadvantaging the farmers. Neither the farmers nor government agencies are able to make necessary adjustments for lack of timely information on what farmers plan to cultivate, or have cultivated. The yield could be affected by various factors including availability of water, weather, and pests. Farmers can make an informed decision on what crop to grow if they have mobile phone-based access to an information system to inquire what others in that region are growing. The information system can provide this information only if most farmers use this system and indicate what crop they plan to grow. Aggregating the information provided by the farmers the information system can also inform the stakeholders what has already been grown for better managing the overall crop production.

4.2 Creating a Better Market Dispersion in the Fishing Industry

For the second area we have chosen the situation described by Jensen [6]. The local fishermen did not have vital information on their market conditions such as the competition, other suppliers and buyers, and were ending up selling at a lower price than they could possibly achieve. With the introduction of mobile phones the fishermen were able to exchange information about the market better and, as a result, the price dispersion was reduced, fish were allocated efficiently across the market, and waste was reduced.

In the above situation, however, every fisherman coming to the shore with a catch of fish has to call a few people on the shore to get some idea about the prevailing market conditions along the shoreline. The fishermen then individually need to aggregate the sketchy information that they have received and decide where they should take their boats to optimize their income. This is cumbersome for fishermen and is prone to errors. This manual information aggregation process can be improved significantly with an automated information aggregation system. This can also lead to many more people along the shoreline being able to provide the information about their need for fish on the day leading to creation of a more accurate picture of the prevailing demand. Such a system can easily be extended to other commodities such as vegetable and staple food. When this happens the fishermen who were mainly the consumers of the aggregated information can also become information producers by informing their needs for vegetables and other commodities. The farmers and merchants on the shore line can now become consumers of this broader source of local information. This scenario is shown in figure 2.

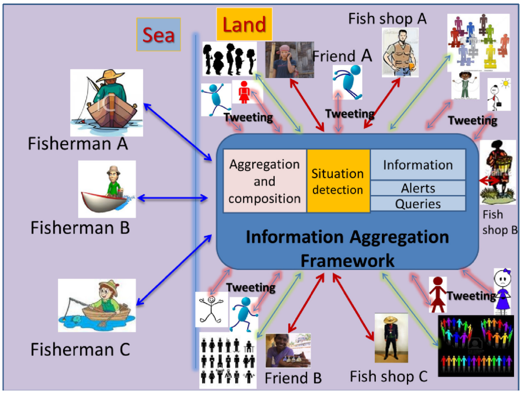


Fig. 2. Creating a better market dispersion through aggregation of micro-information

5 High-Level Conceptual Architecture

To develop MBIS that can address the above needs, it is necessary that the system should able to aggregate the micro-information sent by the users to create a picture of the overall situation and make it available to the users. Also as identified during focus group discussions, users need access to both static and dynamic information. The high-level conceptual architecture for a MBIS for MOP applications is shown in figure 3.

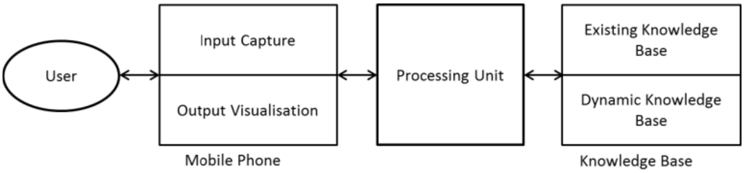


Fig. 3. High-Level Conceptual Architecture

We included Jain's information aggregation framework described in section 3.3 to obtain dynamic knowledge about the prevailing conditions by aggregating micro-information sent by users. Users provide information to the system via a mobile interface. User input may be un-structured or semi structured and in a language the user is familiar with; but the processes in the Knowledge Base needs structured input using a specific language. Also the output from the Knowledge-Base may need to be translated back to the native language of the user. One of the functions of the Processing Unit is to do this conversion. To support different types of user queries, in addition to the dynamic knowledge derived by aggregating micro-information it will also links to existing static knowledge bases.

Our early analysis showed the importance of user empowerment. Thus we next analysed how we can incorporate user empowerment into the above system.

6 User Empowerment

Empowerment is defined as “an intentional, ongoing process centred in the local community, involving mutual respect, critical reflection, caring and group participation, through which people lacking in equal share of valued resources gain greater access to and control over those resources”[18] . Group [19] has defined empowerment as “a process where individuals learn to see a closer correspondence between their goals and sense of how to achieve them, and a relationship between their efforts and life outcomes”.

Both of the above definitions define empowerment as a process which has outcomes. The empowerment theory suggests that actions, activities or structures may be empowering, and the outcome of each process results in a level of being empowered [20]. Applying this general framework to community level analysis, empowerment may refer to collective action to improve the quality of life in a community and to the connections among community organisations and agencies. This can provide the basis to investigate how users can form into communities similar to the “group” concept in current Social Networks. In empowerment theory, empowering processes and their outcomes are clearly defined. Empowering processes for individuals might include learning decision making skills, managing resources and working with others: empowering processes for community might include being able to access government agencies, media and other community resources. Possible outcomes of individuals feeling “empowered” would be situation specific perceived control, critical awareness, skills and proactive behaviours. For communities, empowerment outcomes might include the evidence of pluralism, the existence of organisational coalition and accessible community resources.

7 Proposed Logical Architecture to Support Empowerment

Figure 4 shows the high-level logical architecture of the proposed system.

The processing unit of the system has two parts: pre and post processing and tools. The interface management process converts the structure of the information sent by the users via their mobile phones to the relevant structure that is required by the other parts of the system. User management process manages users by providing relevant login access to the system. This system can receive different types of information and the function of the context analysis process is to identify the type of the information and relay it to the relevant process/s in the system. For example, if it is a query, it will be sent to the query engine or if it is part of a discussion to the discussion tool.

Tools section of the processing unit has some processes that will assist individual and community level empowerment. By participating in discussions and getting involved in community activities, users will learn how to work as a group, make their own decisions that would suit their own environment, share their experiences, create, plan and organise their own events and help one another. The query engine will process the user request for information. Depending on the type of the query, it will be sent either to the dynamic or existing knowledgebase and the user will receive primary and secondary information. For example, the user may have a query about a pest. Depending on the user profile, the user

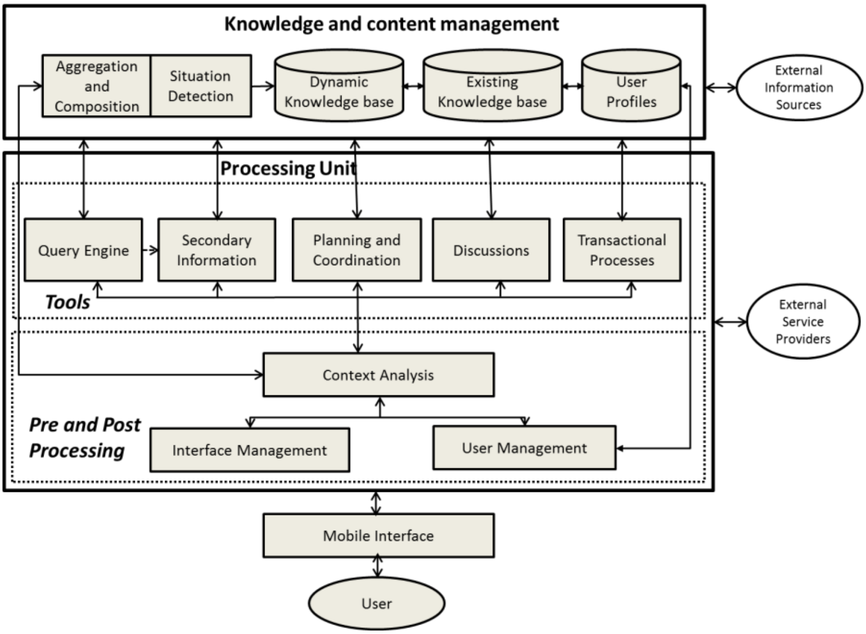


Fig. 4 High level Logical Architecture of the enhanced system

will receive primary information about the pest and also receive links to secondary information such as pest control. This would give the user a choice to learn more about pests and pest control. If the user decides to purchase pest control chemicals, the user can even participate on a discussion forum about this topic before buying and make an informed decision in the end. The transactional process of the tools section will help the users with activities that are related to business transactions and banking processes. Therefore, at an individual level, these processes will help users to learn how to access information, gather new knowledge and make informed decisions.

Dynamic Knowledge about evolving situations is obtained using the aggregation framework described in section 3.3. The existing knowledge base will contain relevant ontologies and links to publically available data sources related to the application area where this system will be deployed.

8 Conclusions, Current and Future Work

There is a great need for developing mobile applications to support the livelihood of people in developing countries that belong to the middle of the pyramid. In this paper, we have presented a high-level logical architecture that can be used to develop these new applications by particularly focusing on a new paradigm known as Social Life Networks. These networks can provide near real time information by aggregating micro information provided by users via various sensors and by using the powerful features such as user empowerment provided by Web 2.0.

The research approach that we used was first to gather all relevant information from literature reviews, focus group meetings and specific technology reviews, then to construct potential application scenarios and develop the architecture to support these applications.

An international collaborative research project has now been started to implement the architecture proposed in this paper [21, 22]. Sri Lanka was selected to trial this application first. An initial application will be developed for the farming domain. The collaborating Universities are University of California in Irvine, University of Salerno in Italy, University of Western Sydney in Australia, University of Macquarie in Australia and University of Colombo in Sri Lanka. This will enable the proposed architecture to be converted into detailed designs, implemented and evaluated. The Italian team is investigating the research issues related to development of the mobile user interface for this application. The Australian team is investigating the research issues related to the development of the processing unit including the query engine and tools for user empowerment. The UC Irvine group is developing the micro information aggregation framework required to implement this architecture. The Sri Lankan team is developing the ontology and knowledgebase for the farming domain within the Sri Lankan context. The Sri Lankan team is also conducting an action research project to study how we can make an intervention to the farming and food distribution life cycle using this system and how it will impact the livelihoods of farmers.

Through these studies we will be able to get a greater understanding of how to develop Social Life Networks to empower people at the middle of the pyramid. The proposed architecture will act as the glue to integrate all the modules to form the overall system.

Acknowledgements. When developing this architecture we received a lot of valuable input and support from researchers, farmers, Government Agencies as well as Non-Government Agencies (NGOS) in Sri Lanka. We also had many interactions with the members of the international collaborating research team. It is too numerous to list them all here. Their names are on the project website http://www.sln4mop.org/. We would like to acknowledge the valuable contribution made by all of them in developing this architecture.

References

1. Cormode, G., Krishnamurthy, B.: Key Differences between Web1.0 and Web2.0. First Monday 13(6) (2008)
2. ITU, Measuring the Information Society: Geneva (2010)
3. Newsroom, G.: Gartner Symposium/ITxpo 2011: Barcelona (2011)
4. Jain, R., Sing, V., Gao, M.: Social Life Networks for Middle of the Pyramid (2011), `http://ngs.ics.uci.edu/whitepapers/SLN_whitepaper%20110212.pdf`
5. Maslow, A.H.: A Theory of Human Motivation. Psychological Review 50, 370–396 (1943)

6. Jensen, R.T.: The Digital Provide: Information (Technology), Market Performance and Welfare in the South Indian Fisheries Sector. Quarterly Journal of Economics 122(3), 879–924 (2007)
7. Aker, J.C., Mbiti, I.M.: Mobile Phones and Economic Development in Africa. Economic Perspectives 24(3), 207–232 (2010)
8. Totty, M.: The 2008 Technology Innovation Awards. The Wall Street Journal (2008)
9. Standage, T.: Mobile Marvels. In: The Economist - A Special Report on Telecoms in Emerging Markets, London (2009)
10. WHO. World Health Statistics (2010), `http://www.who.int/whosis/whostat/2010/en/index.html`
11. Muyinda, P.B., Lubega, J.T., Lynch, K.: Mobile Learning Objects Deployment and Utilization in Developing Countries. International Journal of Computing and ICT Research, Special Issue 4(1), 37–46 (2010)
12. Kumar, A., Tewari, A., Shroff, G.: An Exploratory Study of Unsupervised Mobile Learning in Rural India. In: CHI 2010, Atlanta, Georgia, USA (2010)
13. Power, T., Shrestha, P.: Mobile technologies for (English) language learning: An exploration in the context of Bangladesh. In: IADIS International Conference: Mobile Learning, Porto, Portugal (2010)
14. Ochieng, Z.: Kenya's M-Pesa Now has 9 Million Customers. CIO East Africa Magazine (2010)
15. Agriculture. Department of Agriculture Govenment of Sri Lanka (November 28, 2011), `http://www.agridept.gov.lk/`
16. Singh, V.K., Gao, M., Jain, R.: Social pixels: genesis and evaluation. In: Proceedings of the International Conference on Multimedia, pp. 481–490. ACM, Firenze (2010)
17. Hettiarachchi, S.: Leeks Cultivators Desperate as Price Drops to Record Low. In: Sunday Times, Sri Lanka (2011)
18. Mechanic, D.: Adolescents at risk: New Directions. In: Seventh Annual Conference on Health Policy, Cornell University Medical College (1991)
19. Group, C.E.: Empowerment and Family Support. Networking Bulletin, 1–23 (1989)
20. Zuckerman, E.: Decentralizing the Mobile Phone: A Second ICT4D Revolution? Information Technologies & International Development 6(SE), 99–103 (2010)
21. Jain, R., Sonnen, D.: Social Life Networks. IT Professional 13(5), 8–11 (2011)
22. Ginige, A.: Social Life Networks for the Middle of the Pyramid (2011), `http://www.sln4mop.org//index.php/sln/articles/index/1/3` (cited March 25, 2012)

Towards Quality Monitoring and Evaluation Methodology: Higher Education Case-Study

Olga Cherednichenko and Olga Yangolenko

National Technical University "Kharkiv Polytechnic Institute", Frunze str. 21, 61002, Kharkiv, Ukraine
{marxx75,olga_ya26}@mail.ru

Abstract. Quality monitoring is a continuous function that uses the systematic data collection to provide management with indications of goals achievement. Higher education is considered in this paper. Problems of higher education quality monitoring are discussed. The reference model of quality monitoring and evaluation is suggested. Web mining is suggested as data collection method for higher education quality monitoring system. The architecture of higher education quality monitoring system is based on agent-oriented methodology.

Keywords: quality, monitoring and evaluation, information system, agent-oriented methodology.

1 Introduction

The necessity of education adaptation to the requirements of modern society causes the search of reasonable estimates of education quality and observation of its development processes. The most important condition of education quality improvement is systematic control and analysis of objective data. Monitoring is implemented with the purpose to get objective information about the state of education area and to prevent negative tendencies in its development. The process of education quality management may be efficient only in case of permanent feedback presence, which provides reliable and valid information about the quality of educational system and its parts. But in practice such a feedback is not properly organized.

Education quality assessment is implemented via two groups of approaches. The first one includes approaches based on external assessment. Licensing and certification procedures, universities rating refer to this group. They provide education quality assessment by external organizations or experts. The second group contains approaches based on internal assessment. Surveys, questionnaires, self-assessment are ascribed to this group.

Nowadays, as a rule, modern business management systems include monitoring and evaluation (M&E) subsystem to support reasonable decision-making.

There are two approaches of M&E conducting. The traditional approach known as implementation-focused M&E is oriented on the system inputs, outputs and executed activities [1, 2]. It provides information on administrative and implementation issues.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 120–127, 2013.

Thus the outcomes of the system functioning stay out of consideration. Higher education establishment (HEE) licensing is a good example of implementation-focused M&E. Another example of implementation-focused M&E is accounting the data about universities entrants, students and graduates. This includes information about the level of their knowledge with the help of exams conducting.

The more advanced is results-based M&E which is oriented on goals and results of the system functioning [1, 3]. It is successfully used by financial and coordinating organizations like United Nations Development Program, World Bank and its regional agencies for monitoring and evaluating business projects and policies [2, 3, 4]. It implies collecting and analyzing information about the state of the system to compare how well it is functioning against the expected results. Results-based M&E provides information on development effectiveness issues. It observes the progress toward outcomes.

The goal of this research is monitoring and evaluation information system development. The given paper is devoted to monitoring issues in HEE. Monitoring framework as a component of quality results-based M&E reference model is discussed below as well as the software architecture for its realization.

The rest of this paper is organized in the following way. Section 2 describes elaborated reference model of quality monitoring and evaluation and represents its components. Section 3 is devoted to multi-agent architecture design of monitoring information system. Section 4 presents conclusions and prospect on future work.

2 Reference Model of Quality Monitoring and Evaluation

Education quality monitoring is continuous collection of data on education quality indicators used to provide management of HEE and main stakeholders with indications of the achievement of objectives of education system. The obtained information can be used for reasonable decision-making. Monitoring is always complemented by evaluation process. Results-based M&E system is a powerful tool that management can use to measure outcomes and feed the obtained information back into the process of decision-making.

The given work represents the reference model of the results-based quality M&E (fig.1). The main stakeholder and initiator of M&E is management of organization. Its mission, strategy and plans define goals of organization's functioning and corresponding outcomes that reflect goal achievement.

Based on these goals and outcomes the indicators must be defined. This step is implemented via special quality evidence framework (fig. 2). Experts and analysts should be involved in it. This process requires analysis of goals and outcomes. The substantiation of particular indicators must prove that it really characterizes the given outcome.

Quality monitoring process is based on quality evidence framework and quality monitoring framework. Quality evidence framework proves the essence of particular indicators for quality monitoring and fills those indicators with values (fig. 2). To define the set of indicators the quality model of monitoring object has to be specified. The quality model is influenced by goals and corresponding outcomes of monitoring process.

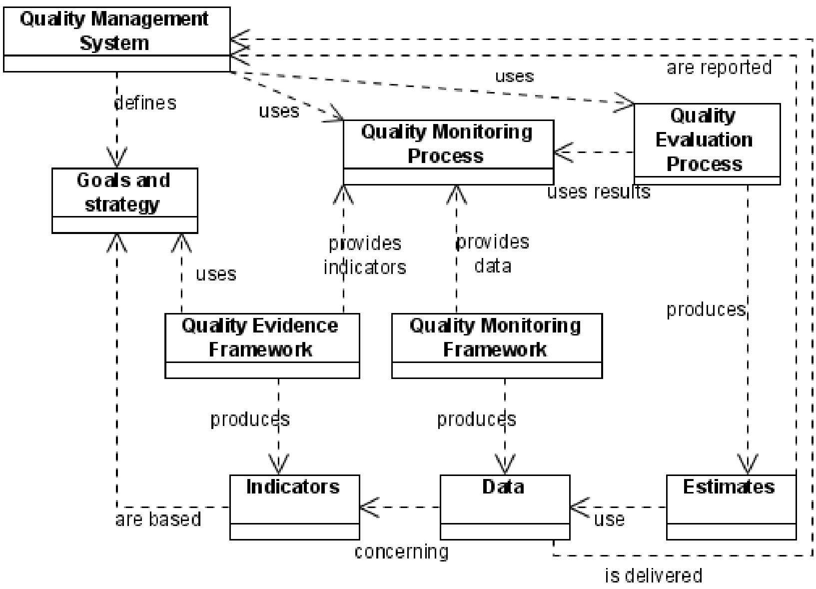


Fig. 1. Reference model of quality M&E

It defines the information which is important for quality management system. Evidence rules are used to conduct reasoning from quality model to indicators. They represent some logic which proves that particular indicators reflect some quality aspects of monitoring object. These rules form the basis for methods of indicators' definition. These methods mostly include experts methods. On the other hand, monitoring object is represented by its model which contains some salient object's features that express its quality. The values of those features are the values of defined indicators. To obtain them the measurement model is used. The procedure of indicators' values calculation is determined by some methods. The most suitable methods include statistical and experts methods.

Quality monitoring framework deals with data which represents the values of indicators (fig. 3). This data can be extracted from definite sources. To collect the data some special methods must be applied. The data collection methods strongly depend on sources of data, as different sources contain information in various forms. The data can be collected by means of people or some automated tools. The data collection can be realized as a part of the whole monitoring IT.

The collected data must be stored somewhere in some appropriate form. Data in the form of paper documents, electronic documents or files can be stored in such data stores as databases or knowledge bases.

The frequency of data collection is defined by monitoring object, its nature and internal features. Frequency is one of the factors that influence the value of costs for monitoring system sustaining.

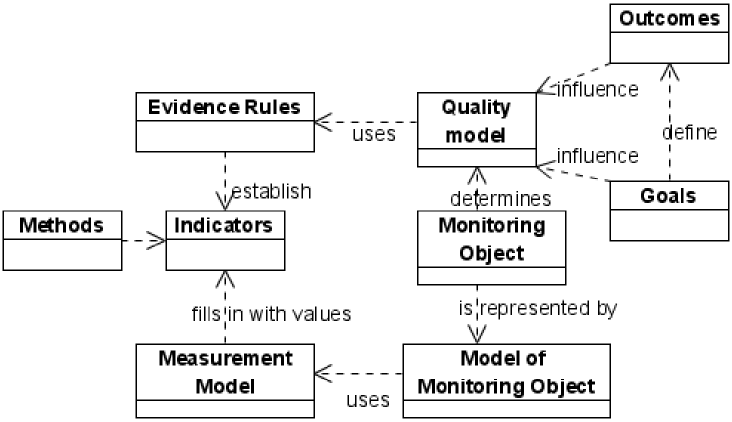


Fig. 2. Quality evidence framework

The collected data must meet the requirements of quality criteria. These include reliability, validity and timeliness of the obtained values of indicators.

Monitoring is strictly connected with data collection methods. They include processing of existing data, various interviews, observations, surveys and so on. All these methods finally require statistical processing of the obtained data. This can be managed by different types of statistical analysis, such as factor, correlation, variance, discriminant analyses [5].

There are various sources of data which can be useful for education monitoring system (fig.4). Among them we can name official reports and records made by management stuff of HEE, for instance, by personnel and practice departments, dean offices. Also this data can be taken from graduates' and students' surveys with the help of special questionnaires that reveal their attitude to education quality of their HEE.

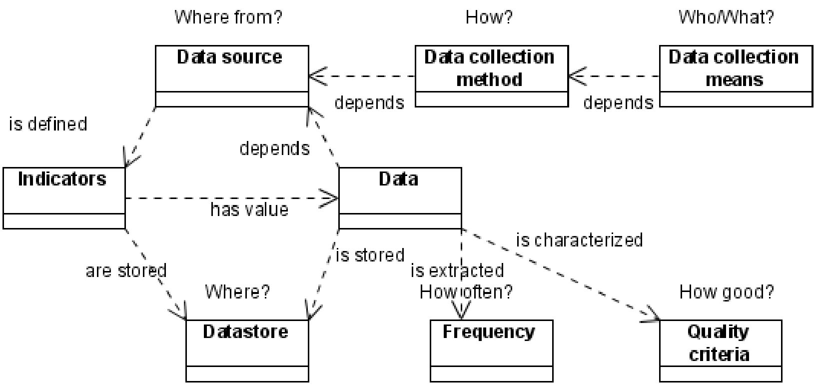


Fig. 3. Quality monitoring framework

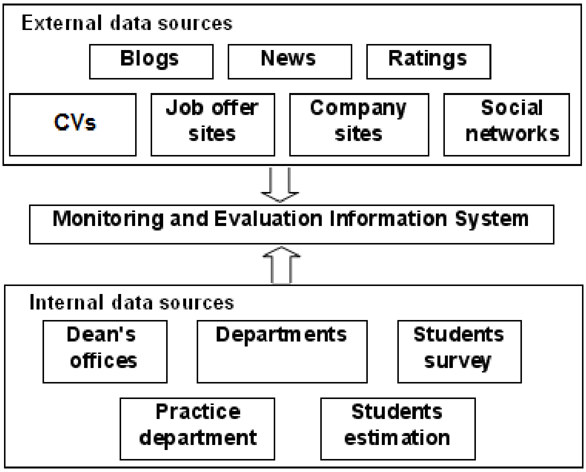


Fig. 4. Data sources

The most convenient source of data for quality monitoring is the results of evaluation of students' knowledge. For example, these include the results of exams passing and current marks.

All of these sources provide data that characterize education quality from inside the HHE. The education quality assessment based on this information will represent the estimate of HEE by itself. Although partially obtained estimates may reflect the education quality from the point of view of external stakeholders. The data from official reports is collected by public authorities, but it is not used for integral education quality assessment from the whole state's point of view. Students and graduates are asked of their perception of education quality, but they are still inside this system and there is no efficient way to find out their opinion over some time when they understand the real value of the obtained knowledge.

There is another source of data that is not taken into account in traditional monitoring systems. The data that characterizes education quality can be gathered from the Internet. The given research considers the Internet as a source of data for monitoring system of higher education quality. Some open information that concerns employment rates can be found there. The sources of the data are banks of CV, social networking sites, personal sites, sites of placement services, job-searching sites with vacancies, enterprise corporate sites. For example, on-line banks of CV contain thousands of CVs where people mention their HEE and working experience. All of the named types of web-sites contain information that indirectly indicates the quality of education obtained in some HEE.

The estimates of education quality that are obtained by processing the data from the Internet seem to be external ones. The problem is to find the targeted web-pages and to get the necessary information from them. This problem can be solved using web mining techniques.

So M&E information system for HEE has to support such data collection methods as reviews of official records, questionnaires and surveys of students and graduates and web mining.

After goals and results of M&E are determined and quality evidence and monitoring frameworks are elaborated (in particular, the quality indicators, the data sources and data collection methods are defined) it is necessary to set baselines. This step provides initial data on indicators. The obtained results are delivered to management for planning of some activities that should be executed to improve the situation. Planning for improvement implies elaborated facilities to achieve some target performance of system. Monitoring and evaluation are performed continuously. The analysis of findings may result in adjustments that should be made to goals and outcomes or indicators. If analysis shows that no adjustment is needed, the management system obtains report.

3 Architecture for Monitoring Framework

In this work we consider the sources of data from the Internet that characterize education quality. This kind of data is located on different web-sites. People describe their HEEs on the personal sites, in social networks, on forums, in blogs, etc. There are different surveys, social researches which results may be placed in some articles on the Internet. Another kind of information can be found in CVs that are available on job-hunting web-sites. The opinion of employers about the education quality of some particular HEE is presented in surveys and on corporate web-sites.

The data described above can differ greatly. It can be structured or not. For example, the data from CVs has more or less formal and common structure while data from blogs is unstructured.

So considering the data search we face two main problems. The first one is finding web pages that may contain the necessary information on education quality. The second one is extracting exactly the targeted data from those web pages. These two types of search require application of different web mining methods and techniques.

The data search is one of the factors that restrict the architecture of monitoring system. Another one is that this system is a part of the management system. It must be integrated into the existing management information system without essential efforts. This affects questions of scalability and portability. So the monitoring system must correspond to the principles of the open system.

Agent-oriented methodology seems to meet our needs at design the monitoring system. This is explained by its advantages comparing to the other approaches.

From software engineering point of view an agent is a long-term, autonomous software entity that adapts its behavior according to environmental changes and interacts with other agents [6, 7, 8]. The agents perceive environment through sensors and act upon it through effectors employing user's knowledge. Three main features of agents are their autonomy, interactivity and learning. So agents can decide by themselves how to act in order to achieve its objective. And the agents can communicate with each other similar to social activities. Learning as a feature is associated with intelligent agent from artificial intelligence domain. Agent-oriented approach combines the features of object-oriented and component-based methodologies, but providing possibilities to realize all the defined agents' properties. Agent-oriented methodology allows implementing multi-agent systems (MAS) which are composed of cooperative agents that interact with each other to achieve individual or common goal.

In the given work we suggest the architecture of HEE quality M&E information system on the basis of MAS (fig. 5). There are several kinds of agents. Web-resources searching agent is responsible for finding information from the external data sources. Screen scrapping agent searches for necessary data at the targeted web pages. Data analysis agent provides the quality indicators with values. Data cleaning agent supports initial data pre-processing and eliminates the repeated or inconsistent data.

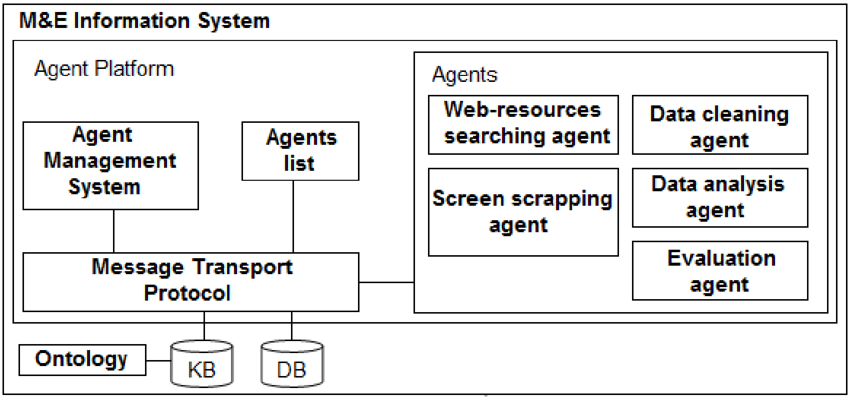


Fig. 5. M&E information system architecture

The main goal of the evaluation agent is to give the estimates of education quality. The agents communicate with each other via message transport protocol. Agents' work is coordinated and controlled by agent management system. Agents list is used to inform each agent about its environment, i.e. all existing agents and their functions and goals. Ontology is necessary to provide agents with common description of universe of discourse. Data necessary for functioning of the M&E IS should be stored at knowledge base and database.

4 Conclusions and Future Work

Under the current conditions management of higher education establishment can be hardly imagined without the quality management compound. The given research represents initial steps for creating quality monitoring and evaluation information system targeted at management improvement. The suggested reference model of quality M&E implies realization of quality evidence and monitoring frameworks.

In this paper we pay much attention to monitoring issues. Among the whole set of monitoring tasks it requires defining data sources and data collection methods. For higher education quality the internal and external data sources are distinguished. The emphasis is made on external data sources as they are not considered in traditional monitoring activities.

The agent-based architecture of M&E information system is proposed. Each agent deals with particular part of the system functionality. The intelligence property of agents is especially useful for solving searching tasks.

The further researches will be devoted to realization of data searching, particularly by means of web mining in case of external resources. The questions of obtained data reliability and validity are also the subject of future work.

References

1. Kusek, J.Z., Rist, R.C.: Ten Steps to a Results-Based Monitoring and Evaluation System: a Handbook for Development Practitioners. The World Bank, Washington, DC (2004)
2. United Nations Development Programme. Handbook on Monitoring and Evaluating for Results. Evaluation Office, New York (2002)
3. Farrel, G.M.: Results-Based Monitoring and Evaluation at the Commonwealth of Learning: A Handbook. Commonwealth of Learning, Vancouver (2009)
4. Campbell, B.R., Lockie, E., Yan, X.: People's Republic of China: Developing a Result-Based Monitoring and Evaluation System for Key Projects. Report by Asian Development Bank (2007)
5. Ross, S.M.: Introduction to Probability and Statistics for Engineers and Scientists. Elsevier Academic Press, London (2004)
6. Symeonidis, A.L., Mitkas, P.A.: Agent Intelligence Through Data Mining (Multiagent Systems, Artificial Societies, and Simulated Organizations). Springer-Verlag New York, Inc., Secaucus (2005)
7. Henderson-Seller, B., Giorgini, P.: Agent-Oriented Methodologies. Idea Group Publishing, London (2005)
8. Rouff, C.A., Hinchey, M., Rash, J., Truszkowski, W., Gordon-Spears, D.: Agent Technology from a Formal Perspective. Springer, London (2006)

Validating OntoElect Methodology in Refining ICTERI Scope Ontology

Olga Tatarintseva, Yuriy Borue, and Vadim Ermolayev

Department of IT, Zaporozhye National University,
66 Zhukovskogo st., 69600 Zaporozhye, Ukraine
tatarintseva@znu.edu.ua, yura.borue@gmail.com,
vadim@ermolayev.com

Abstract. This research work is focused on the experimental validation of our OntoElect approach to ontology engineering in the case study of iterative refinement of the ICTERI Scope Ontology. OntoElect has been used for evaluating the commitment of the domain knowledge stakeholders to the ontological offering measured as positive and negative votes. The analysis of the measures allowed us answering the questions about the fitness of the ontological offering to the implicit requirements of the stakeholders as well as the completeness of the domain model with respect to their implicit expectations. The paper briefly presents the idea of OntoElect as a socially inspired approach for ontology engineering. It further describes the objectives and the set-up of our evaluation experiment. Finally it presents experimental results.

Keywords: ICTERI Scope Ontology, OntoElect, ontology engineering, domain knowledge stakeholder, socially inspired approach, evaluation, refinement.

1 Introduction

One of key issues in developing and refining ontologies is making them fit to the changing requirements and attitudes of the subject domain stakeholders which constitute their evolving commitments to a large extent. Available knowledge engineering methodologies lack in providing a robust routine for coping with this issue fully adequately. The major obstacle is not even in capturing stakeholder commitments as change requirements, but in assuring that the commitments are expressed and made known to knowledge engineers. Subject experts in a domain often consider corresponding requests as an additional overhead for them – not really providing an aid in performing their core professional activities. Hence, finding a way of motivating subject experts to be actively involved in assessing the offered ontology with respect to the fitness to their requirements is important for increasing the commitment of the stakeholders to the resulting knowledge representation artifact.

As suggested in our previous publication [1], a promising approach could be organizing ontology elections – in which different “ontology offerings” compete for the commitment of the pool of the relevant domain knowledge stakeholders who are considered the “electorate”. The problem of motivating the electorate to vote actively

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 128–139, 2013.

remains however a vital one – very similarly to the political elections. In this paper we report how this motivation problem is attempted to be solved in one particular case study – the refinement of ICTERI Scope Ontology[1]. We also assess if the analysis of the opinions of the domain knowledge stakeholders could yield effective results in refining an existing "seed" ontology offering.

The paper is structured as follows. Section 2 briefly overviews the related work in knowledge engineering methodologies with stakeholder community involvement – in particular in the part of motivating subject experts being active in validating ontologies with respect to their individual or group requirements. Section 3 describes briefly our OntoElect approach and the Case Study on the evaluation of the ICTERI Scope Ontology for its further refinement. Section 4 presents the set-up of the evaluation experiment as well as the developed software tools. Section 5 reports the experimental results. Finally, Section 6 draws some conclusions and presents our plans for future work.

2 Related Work

One of the possible ways to check if a conceptualization of a domain is correct and complete is to evaluate the model against the outcome of domain knowledge stakeholders' interpretation of the meaning of the representative set of relevant documents. From the other hand, the labels or comments the stakeholders put on these documents or resources may be effectively used to infer the conceptualization. Such labels are often denoted as tags or annotations.

If web resources of different modalities are thought of as a representative set of data we find ourselves in the exploding field of collaborative or social tagging and annotation – a substantially characteristic part of the Web 2.0 phenomenon. A good survey of the field of social tagging is offered by Gupta et al in [2] where the use of tags for different purposes and associated shortcomings is also analysed. The shortcomings effecting in low quality are: (i) lack of expressiveness; (ii) lack of correctness; (iii) ambiguity; (iv) lack of uniform coverage. The major reason is that the taggers do not use the terms of a consensual domain model in their activity.

Semantic annotation and tagging approaches further refine social tagging techniques by offering the collections of terms that are taken from taxonomies, folksonomies, or thesauri [3]. However, backbone knowledge representations have to be obtained before semantic annotation may be undertaken. Developing those remains the work for ontology engineers and is not regarded as a task for non-specialist users.

Hybrid approaches for collaborative tagging and annotation aiming at the enrichment of seed knowledge representations by a user community also reported, for example in [4]. The weaknesses of traditional social tagging systems are attempted to be overcome by combining the best features of the Social and Semantic Webs. Unfortunately the problem

[1] ICTERI Scope Ontology is the ontology describing the scope of ICTERI conference series (`http://icteri.org/`). The development of this ontology has been initiated at ICTERI 2011 and resulted in the Wiki-based reference resource (`http://isrg.kit.znu.edu.ua/icteriwiki/`) for the authors who submit to ICTERI. So, the authors of ICTERI submissions are our subject domain knowledge stakeholders.

of motivating users actively taking part in annotating resources remains open even in the reported advanced collaborative semantic tagging systems.

As mentioned by many authors, good ontologies should match consensual interpretations of domain knowledge by the representative set of domain knowledge stakeholders as closely and completely as possible. Therefore, ontology engineering in any form is a process that has to involve as many domain experts as affordable. Increasing numbers of people are spending their time using Web 2.0 applications. Remarkably, it is totally on the contrary for ontology development. To the best of our knowledge nobody reported about involving very big user groups in creating knowledge representations so far. As recognized by many experts in Semantic Technologies one of the possible reasons is that traditional ontology engineering methodologies detach the effort from the benefits (c.f. [4]) hence de-motivating the involvement of those people whose interpretations of the domain are critically required.

One of the promising approaches for motivating more people taking part in developing or refining ontologies is offering a social software tool or a game with a purpose to a group of intended users. Ontology creation can be implicitly embedded in such social software, namely social networking portals, where ontology items would be created, evolved, and validated implicitly in the background, while users simply provide information to a portal for sharing content and communicating with other users [5]. Games with a purpose have been used mainly for collaborative tagging of resources having different modalities – for example images [6], music [7]. Gaming approach has also been tried for inferring human intentions from their recorded actions (Common Consensus game, [8]) and for evaluating how well commonsense facts fit to the interpretations of random users (FACTory Game by Cycorp, http://game.cyc.com/). Several game scenarios have been developed [4] for ontology building and refinement, ontology matching, annotating content using lightweight ontologies. Even though ontology backed-up social networking portals and games with a purpose differ in relation to the users' motivation to contribute, they are in compliance to our OntoElect approach ([1] and next Section). Both approaches offer possibilities to identify whenever users start to agree on and share commitment to certain ontological items.

The positive features that are common to all these bits of related work are: (i) the pattern of user involvement adopted from Web 2.0 is used to motivate people take part; (ii) all games with a purpose hide their purpose under the gaming scenario – offering fun or other immaterial incentives in reward for providing useful results; (iii) the scenarios are designed in a way to assist in structuring the pool of players by their reputation.

There is also an inherent shortcoming to this approach – knowledge representations or mark-ups that are crafted by non-specialists can only be lightweight. Otherwise the overhead for ramping-up the players would overweigh all potential incentives. Moreover, though resulting ontology fragments are aligned with consensual user interpretations, it is hard to ensure that the quality and coherence of those fragments is sufficient. Therefore, a joint motivated involvement of ontology engineering professionals, subject experts, and a sufficiently big group of intended users with domain knowledge and expertise is required.

Motivations are naturally denoted as "…reasons individuals have for behaving in a given manner in a given situation" (c.f. [10]). "They exist as part of one's goal structures, one's beliefs about what is important, and they determine whether or not one will engage in a given pursuit'' (c.f. [11]). Broadly, two types of motivation are distinguished – intrinsic and extrinsic. Intrinsically motivated subjects act for their own sake, because enjoy acting or assess the outcomes as important for themselves – e.g. [12]. Extrinsic motivation is driven by a desire of getting rewards – from the others; or avoiding punishment. Subjects motivated extrinsically focus on receiving the approvals – like judgments by fellow colleagues and peers – e.g. [11].

In our settings, for the stakeholders who own the knowledge of the subject domain, a combination of both motivation strands is sought as valid. From one hand, the authors of ICTERI papers are interested in possessing the ontology containing the terms best and most fully describing their own research subfield in the domain – which constitutes their intrinsic motivation. From the other hand, they are extrinsically eager to have a clear terminological representation – and therefore a better visibility and increasing reputation of their subfield within the community.

3 OntoElect and Ontology Refinement Case Study

Our OntoElect approach for involving domain knowledge stakeholders actively in the development and ownership of ontologies describing their domain is inspired by the observations of *public election campaigns*. Indeed, the desired outcome of an election campaign for each candidate is to gain as much commitment of the electorate as possible. Such a commitment is measured adequately by the number of votes. The candidates compete for the votes by presenting their programmes, making coalitions, taking part in public debates – proving that his or her programme is the best match to the expectations of the majority of electorate.

In the context of ontology engineering alternative ontology offerings for the same subject domain could be treated as election candidates. Each candidate ontology offered in a, so to say, *ontology election campaign* may be evaluated compared to the other candidates by the ability to answer the competency questions of the electorate. The more competence in answering the requirements of the electorate is demonstrated by the ontology, the more commitment it is potentially able to gain. The members of the electorate are the intended users in the domain. More details on how the commitment of the electorate is measured and accounted as votes in OntoElect are given in our earlier paper [1].

The case study of refining the ICTERI Scope Ontology has been chosen to validate the OntoElect approach in a controlled iterative ontology engineering experiment. Initial ontology offerings for the four thematic areas of ICTERI scope have been offered at ICTERI 2011. The participants of this conference have been offered to check if these ontology fragments presented as UML class diagrams fit well to their interpretations of the conference scope. A simple questionnaire has been offered for filing in at the conference. Unfortunately the response to the questionnaire was very weak as the participants were not well motivated to spend their conference time in assessing the offered UML models. So we decided to simulate the opinions of our electorate by checking how well could the papers accepted to ICTERI 2011 be annotated by the concepts of the provided ontological offerings. These concepts are further referred to as the ICTERI Key Terms.

For that all the abstracts of all accepted papers have been manually annotated. The annotations have been done using the available most specific ICTERI Key Terms where possible. In several cases we also had to add the Missing Key Terms. For example we added missing terms when only too high-level ICTERI Key Terms fitted for a particular paper, or the topic of this paper was not at all covered by the ontological offerings.

After the semantic annotation of the available pool of papers has been accomplished we measured the votes of the electorate with respect to the particular semantic contexts [13] within our initial ontological offerings. One use of a particular ICTERI Key Term has been considered as one vote for the semantic context having this term as an anchor. The votes have further been normalized as frequencies of use. This measurement allowed us to detect the most popular semantic contexts and circumscribe the part of the ontological model which has been most frequently demanded for annotating the pool of ICTERI papers.

Another important characteristic that we assessed was the completeness of the set of initial ontological offerings. We measured the frequencies of use of the Missing Key Terms as votes very similarly to the previous case. These votes gave us important indications about the necessary extensions and refinements of the ICTERI Scope ontology.

The collection of votes for the available and missing Key Terms was not considered as fully unbiased as we annotated the papers ourselves. So, to lower the level of bias we paid a look also to the keywords freely chosen by the authors for their papers. The keywords have been collected and the frequencies of their use have been analysed for reinforcing our assessment of the completeness of the ICTERI ontology.

4 Experimental Setup and Developed Tool

A controlled experiment has been set up to evaluate the initial ontological offerings of the ICTERI Scope Ontology. We start with presenting the objectives and outlining the metrics related to these objectives. We then present our experimental workflow and the software tools that have been developed for partially automating the required activities. We also present the resource for the ICTERI community that has been elaborated in our case study.

Our measurable objectives for the experiment have been formulated as follows:

- Does the ontology fit to the requirements of the subject experts in the domain?

The fitness of the ontological offering is measured as a ratio of the average frequency of use of the available Key Terms (positive votes) to the similar for the missing Key Terms (negative votes). Special attention is paid to the freely chosen key words that are identical to the available Key Terms. Those are considered as extra positive votes for the semantic context of the Key Term.

- Is there a particular part in the ontology that is the most important for the stakeholders?

The importance of an ontology fragment comprising particular concepts is measured as the frequency of use of these concepts (positive votes). Fragments of different importance are also presented as percentiles.

- Is there a part in the ontology that could be dropped as the stakeholders do not really require it?

Similarly to importance, these ontology fragments are outlined using low frequency of use percentiles.

- What would be a most valuable addition to the ontology that will substantially improve stakeholders' commitment to it?

The papers have been annotated using missing concepts and freely chosen keywords. Those missing concepts that are frequently used form the core of this effective extension. If some of the keywords are also used frequently by the authors they may become good candidates for the inclusion in the effective extension as well. Special attention is paid to the freely chosen key words that are identical to the missing Key Terms. Those will reinforce the votes on the addition to the ontology.

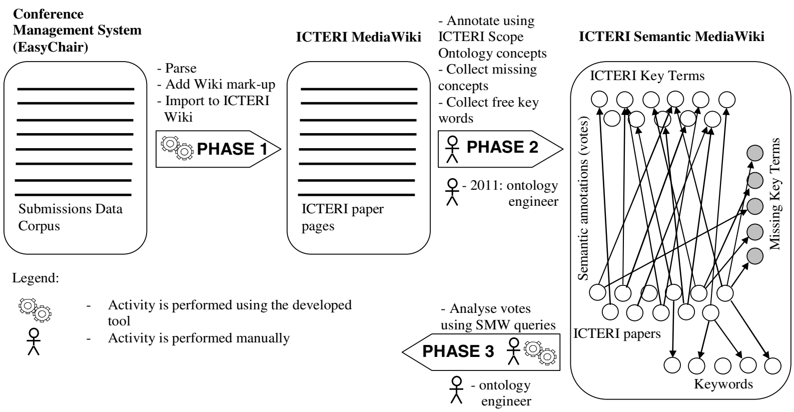


Fig. 1. Workflow for processing ICTERI papers and collecting stakeholders' votes

The flow of activities has been organized in three consecutive phases as presented in Fig. 1.

In phase 1 we have extracted the semi-structured information about the papers accepted for ICTERI 2011 and transformed these into the collection of paper articles in the ICTERI Wiki. The information for each paper has been: (i) extracted from the EasyChair installation of the Conference Management System; (ii) parsed for proper structuring; and (iii) transformed to Wiki styled marked-up representation using

XSLT. The processing pipeline of the tool developed for information extraction, parsing, and transformation (Paper Importer) is presented in Fig. 2.

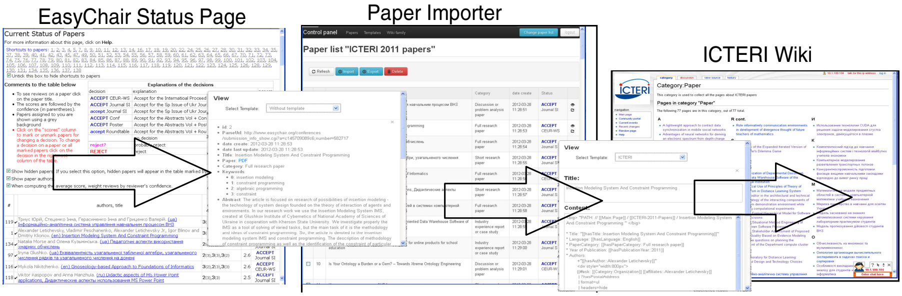


Fig. 2. Paper import processing pipeline (Phase 1 in Fig. 1)

In Phase 2 we performed manual annotation of the imported papers using the concepts of the ICTERI Scope Ontology as the available Key Terms. The most specific relevant Key Terms have been selected in order to make the annotation as precise as possible and the boundaries of the related semantic contexts narrower. For making the selection of the terms easier we have developed a navigable representation of the ontological offerings for the ICTERI Scope Ontology[2].

All the concepts of ICTERI Ontology have been presented in three different ways:

(1) In the navigable UML class diagram at the ICTERI-Terms wiki page
(2) In the alphabetic list of concepts below the corresponding UML class diagram in the same ICTERI-Terms wiki page
(3) And in a separate Concept page reachable in a click from the interactive UML class diagram that gives a definition and a detailed description of the Concept

In some cases we did not find appropriate Key Terms in the available ontological offering, so the missing concepts have been added to the corresponding Key Term annotation in the paper page using Semantic MediaWiki tags.

As a result of this Phase the semantic relationships between the pages of `Category:Paper` and the pages of `Category:Concept` have been specified as semantic properties. These semantic properties allowed us receive all the measurements planned for the evaluation experiment. These measurements have been done using different Semantic MediaWiki queries in Phase 3. For example, the tag cloud for the frequency of use of the Key Terms of ICTERI ontology (Fig. 3) has been elaborated using the following inline query:

```
{{#ask: [[hasKeyTerm::+]]
  | ?hasKeyTerm
  | format=tagcloud
  | mainlabel=-
}}
```

[2] Please refer to `http://isrg.kit.znu.edu.ua/icteriwiki/index.php/ICTERI-Terms`

5 Experimental Results and Discussion

In this section we present and discuss the results of our experiment. This discussion is structured alone the measurable objectives of the experiment explained in Section 4. For obtaining the basic data for building all the necessary estimations we computed the frequencies of use of the available Key Terms (Fig. 3), missing Key Terms (Fig. 4), and freely chosen keywords (Fig. 5).

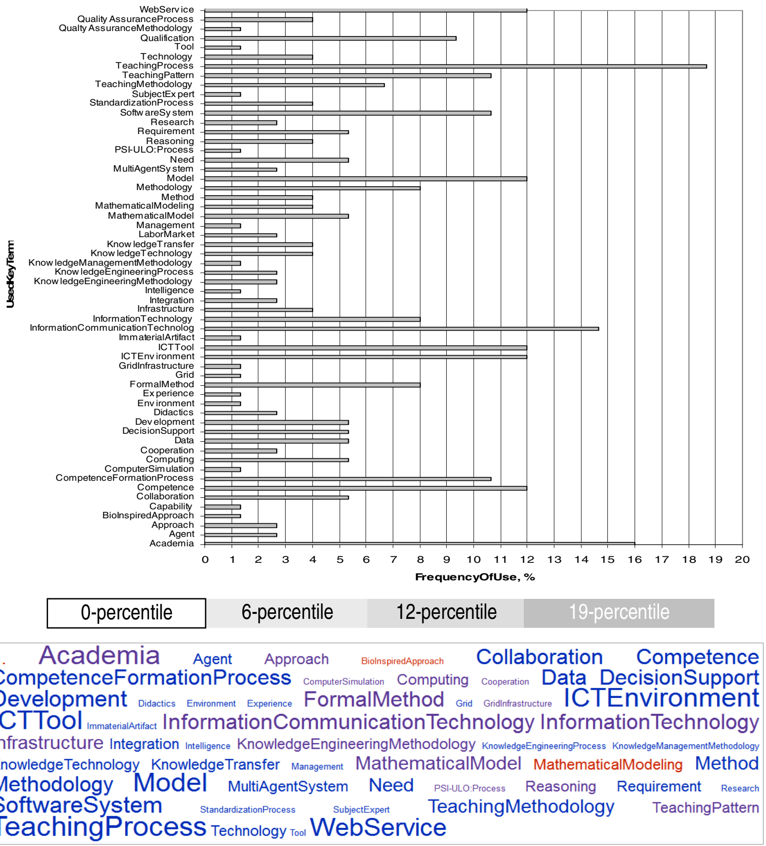


Fig. 3. Frequency of use of the available Key Terms. The diagram at the top pictures the distribution of the frequencies of using the available Key Terms for annotating ICTERI papers (in percentage to the overall No of the processed papers). The Diagram in the bottom of the figure is based on the same frequency data, but provides an indication of the popularity of the Key Terms for our domain knowledge stakeholders in a form of a tag cloud (snapshot made at the ICTERI Wiki).

Using these data interpreted as different kinds of domain knowledge stakeholders' votes we were able to answer the objectives of our experiment in a quantitative manner.

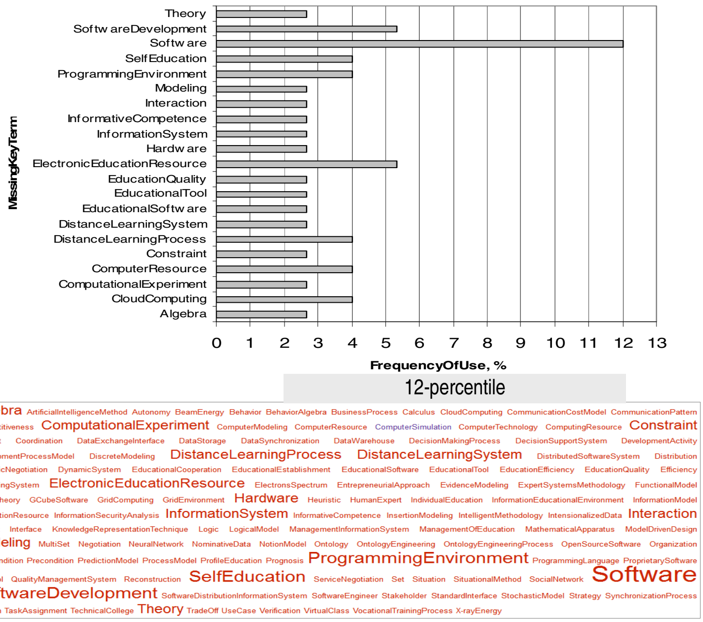


Fig. 4. Frequency of use of the missing Key Terms. Only those missing Key Terms that have been used for annotating more than one paper are included in the diagram at the top. The diagram at the bottom indicates the popularity of all missing Key Terms in the form of a tag cloud (snapshot made at the ICTERI Wiki).

First, we assessed the fitness of the available ontological offerings to the requirements of the domain knowledge stakeholders. It has been considered that the semantic context of a concept in the ontological offering receives a positive vote if the corresponding available Key Term is used for annotating a paper. The satisfaction of all the stakeholders by the ontological offering was modelled as the average frequency of use of the available Key Terms $fi_{+} = 5.29$. Similarly, one use of a missing Key Term was considered a negative vote and the dissatisfaction of the stakeholders was computed as an average frequency of use of the missing Key Terms $fi_{-} = 1.73$. The overall fitness estimation was computed as a ratio of the satisfaction factor to the sum of satisfaction and dissatisfaction factors $fi = fi_{+} / (fi_{+} + fi_{-}) = 0.753$. This value could be interpreted as the authors of ICTERI 2011 papers were approximately 75 per cent happy with the available ontological offering of ICTERI Scope Ontology.

Secondly we circumscribed the part of the ontological offering that is important for the stakeholders. For that we looked at the groups of the available Key Terms with different frequency of use percentiles – as pictured in Fig. 3. Each concept of the ontology has received a weight equal to its individual frequency of use proportional to the number of papers annotated by the corresponding Key Term. Consequently, the backbone taxonomy of the ICTERI Scope Ontology[3] has been built based on the concepts belonging to 6-, 12-, and 19-percentiles (58 concepts. Additionally all the super-classes of the non-zero-percentile concepts (14) were included to preserve the subsumption structure of the ontological offering. The part of the ontological offering represented by the semantic contexts of the concepts included in the backbone taxonomy is regarded as important for the domain knowledge stakeholders as it covers all their positive votes.

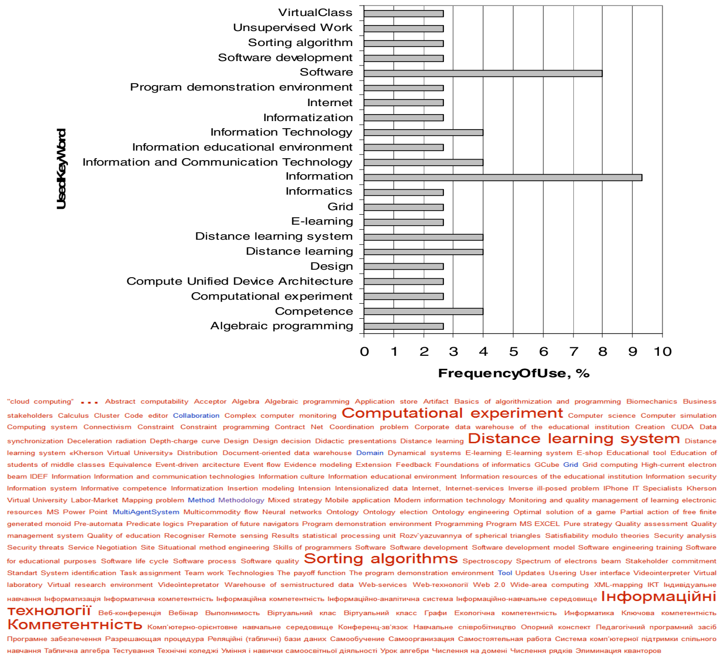


Fig. 5. Frequency of use of the freely chosen key words. Only those key words or phrases that have been used for annotating more than one paper by their authors are included in the diagram at the top. The key words have been pre-processed before building the diagram: (i) those initially given in Ukrainian or Russian were translated to English; (ii) semantically identical were merged. The diagram at the bottom indicates the popularity of all original key words in all three languages in the form of a tag cloud (snapshot made at the ICTERI Wiki).

It is worth noting that in fact the important part of the ontology also comprises all the concepts that are related to the non-zero-percentile concepts by binary associations (additionally 8 concepts). Those have been included for preserving all the properties of the basic concepts. The gradation of importance of different sub-contexts in the backbone taxonomy has been indicated by assigning 6-, 12-, and 19-percentile colours to the corresponding contexts. So, it could be estimated that the commitment of the stakeholders to the subsumption hierarchies comprising the concepts of the 12- and 19-percentile is considerable – pointing to the most important part of the ICTERI

[3] Please refer to `http://isrg.kit.znu.edu.ua/icteriwiki/index.php/ICTERI-Backbone-Taxonomy`

Scope Ontology. The proportion of the number of concepts in the important part to the overall number of concepts in the ontological offering is 0.71.

The part of the ontological offering that presented no value to the pool of our stakeholders has been outlined as a symmetric difference to the important part.

Finally the estimation of the most valuable addition to the ontology has been made based on the analysis of the missing Key Terms and freely chosen keywords. Straightforwardly, the addition compensating all the negative votes of the stakeholders would have been the one based on the set of missing Key Terms. However, the influence of the vast majority of them – used for annotating just one paper – is very low.

So, our estimation is that adding the missing Key Terms of 2-percentile (top diagram in Fig.4) would be most valuable for the ontology. More valuable additions are indicated by the most frequently used key words chosen freely by the stakeholders (top diagram in Fig. 5).

6 Concluding Remarks and Future Work

The objective of the presented research was to validate the OntoElect approach to ontology engineering in the case study of iterative refinement of the ICTERI Scope Ontology. The approach has been used for evaluating the commitment of the domain knowledge stakeholders to the ontological offering measured as positive and negative votes. The analysis of the measures allowed us answering the questions about the fitness of the ontological offering to the implicit requirements of the stakeholders as well as the completeness of the domain model with respect to their implicit expectations.

The metaphor of votes has proved to be useful as it allowed answering the objectives of ontology evaluations in a quantitative manner and suggesting the refinements of the ontology in the next iteration of its development in a grounded and explicit way.

It has to be stated that the presented research is in its intermediate stage and several important aspects need to be elaborated more profoundly. These aspects outline the directions of our future work.

Firstly, the document corpus used so far is still too small in size for assuring reliable judgements about the opinion of the community. For improving that the collection of ICTERI papers will be continued and further iterations of ontology refinement will be undertaken using the extended pool of papers.

Secondly, the quality of annotations based on the manual knowledge extraction from paper abstracts is estimated to be not very high. To improve on that, we plan exploring available supervised and semi-supervised knowledge extraction techniques and applying them to semantically annotate the full texts of papers.

Thirdly, the problem of motivating domain knowledge stakeholders still remains a challenge. Currently we pretended to infer their commitments based on extracting their votes from their papers. For the next iteration we already try a prescriptive approach – it has been requested that the submissions to ICTERI 2012 contain Key Term annotations in addition to the freely chosen key words. The problem of incentivising the stakeholders to take their active part voluntarily still remains unsolved and requires further research.

Fourthly, the metrics we used for evaluating the fitness and the completeness of ontological offerings are still quite simple and do not account for several important aspects. One of these aspects is the account for the influence of a vote on the semantic context of a concept causing possible propagation of a vote to the adjacent concepts depending on the nature of corresponding relationships. Refining metrics and proving the adequacy of these refinements in the context of our work is planned for the future.

Finally, our processing pipeline in its current state (Section 4) still comprises many manual steps. Furthermore, the developed software tool is not integrated in the ontology engineering infrastructure. We plan spending some effort in this respect – namely for developing the analytic functionality for Phase 3 and further integrating all the tools in an ontology engineering environment in the mid term.

References

1. Tatarintseva, O., Ermolayev, V., Fensel, A.: Is Your Ontology a Burden or a Gem? – Towards Xtreme Ontology Engineering. In: Ermolayev, V., et al. (eds.) Proc. 7-th Int. Conf. ICTERI 2011, Kherson, Ukraine, May 4-7. CEUR-WS.org, vol. 716, pp. 65–81 (2011) ISSN 1613-0073
2. Gupta, M., Li, R., Yin, Z., Han, J.: Survey on Social Tagging Techniques. SIGKDD Explorations 12(1), 58–72 (2010)
3. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Science. Services and Agents on the World Wide Web 4(1), 14–28 (2006)
4. Hunter, J., Khan, I., Gewrber, A.: HarvANA – Harvesting Community Tags to Enrich Collection Metadata. In: Paepcke, A., Borbiha, J., Naaman, M. (eds.) 8th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 147–156. ACM, New York (2008)
5. Siorpaes, K., Hepp, M.: Games with a Purpose for the Semantic Web. IEEE Intelligent Systems 23(3), 50–60 (2008)
6. Zhdanova, A.V.: Community-driven Ontology Construction in Social Networking Portals. Web Intelli. and Agent Sys. 6(1), 93–121 (2008)
7. Von Ahn, L.: Games with a Purpose. Computer 29(6), 92–94 (2006)
8. Law, E., et al.: Tagatune. In: Proc. Int. Conf. Music Information Retrieval, pp. 361–364. Austrian Computer Soc., Ismir (2007)
9. Lieberman, H., Smith, D., Teeters, A.: Common Consensus: A Web-Based Game or Collecting Commonsense Goals. In: Proc. Workshop Common Sense for Intelligent Interfaces, ACM Conf. Intelligent User Interfaces. ACM Press (2007)
10. Middleton, J.A., Photini, A.: Spanias: Motivation for Achievement in Mathematics: Findings, Generalizations, and Criticisms of the Research. J. Research in Mathematics Education 30(1), 65–88 (1999)
11. Ames, C.: Classrooms: Goals, structures, and student motivation. J. Educational Psychology 84, 261–271 (1992)
12. Middleton, J.A.: A study of intrinsic motivation in the mathematics classroom: A personal constructs approach. J. Research in Mathematics Education 26, 254–279 (1995)
13. Ermolayev, V., Ruiz, C., Tilly, M., Jentzsch, E., Gomez-Perez, J.M.: A context model for knowledge workers. In: CIAO 2010. CEUR-WS.org, vol. 626 (2010) ISSN 1613-0073

Composition-Nominative Logics in Rigorous Development of Software Systems

Mykola S. Nikitchenko and Valentyn G. Tymofieiev

Department of Theory and Technology of Programming
Taras Shevchenko National University of Kyiv
64, Volodymyrska Street, 01601 Kyiv, Ukraine
nikitchenko@unicyb.kiev.ua,
tvalentyn@univ.kiev.ua

Abstract. We provide motivation for composition-nominative approach and its main underlying principles. We give a brief introduction to the hierarchy of composition-nominative logics (CNL) and discuss possible applications of these logics in formal approaches for software development. CNL are algebra-based logics of partial predicates and functions. They are constructed in a semantic-syntactic style on the methodological basis, which is common with programming. Main attention in the paper is paid to the first-order CNL. The satisfiability problem for the first-order CNL is addressed. We show the reduction of the problem in hand to the satisfiability problem for classical first-order logic with function symbols. The reduction proposed requires extension of logic language and logic models with an infinite number of unessential variables. The method presented in the paper enables one to use existent satisfiability checking algorithms and tools for solving instances of the CNL satisfiability problem. Thus, many standard techniques developed for classical logics can be applied in CNL-based approaches for program reasoning.

Keywords: Composition-nominative approach, formal methods, partial predicate, partial logic, first-order logic, satisfiability, validity.

1 Introduction

There is a variety of proposed approaches for software development ranging from the early ones as waterfall model to more recent ones as agile methodology. The fact that so many different approaches appeared confirms that development of industrial software systems is a very difficult problem. One of the difficulties is that software engineering process is error-prone, and errors can appear even on early stages of development. As a way to increase software reliability and clarify the notion of software correctness a number of mathematically based approaches for software (and hardware) development appeared. These approaches are usually referred to as formal methods.

Formal methods are grounded on different branches of mathematical logic, set theory, automata theory, formal languages theory, and other fundamental areas of mathematic. Depending on the degree of formality, formal methods can be classified in three styles [1]:

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 140–151, 2013.

- formal specification only, when formal techniques are applied only on specification stage;
- formal specification and rigorous development, when formal techniques are applied on specification and on development stages so that one proceeds from more abstract to more concrete specifications, but does not prove the refinement relations between them formally (or proves only a few of them);
- formal specifications and formal development, when formality is used both during specification and development stages, and the refinement relations between abstract and concrete specifications (programs) are formally proved.

It is difficult to achieve the degree of absolute formality; therefore rigorous development is more realistic.

Usage of formal methods also leads to considerable increase of expenses for system development. But the common view is that formal methods still have to be used in case of safety-critical systems development.

Rigorous methods are based on formal program specifications, further referred to as *program model*s. Such models should reflect features as:

- partiality: programs may be undefined on some input data;
- usage of complex system of data types: arrays, hash-tables, linked lists, semi-structured data, etc.;
- non-determinism: programs can evaluate to different results on the same input data;
- possibility of transformation of abstract specifications (programs) to more concrete ones.

We take into consideration these specifics when constructing our formalism for program description. The approach proposed is called *composition-nominative approach* (CNA) [2]. It is based on principles of development from abstract to concrete, compositionality, and nominativity. Compositionality means that properties of complex systems (programs) are specified by properties of their components. Nominativity (from Latin *nomen* – name) emphasizes the importance of naming relations in program specification and construction. As a result we obtain a hierarchy of *composition-nominative program models* (CNPM).

Application of formal methods based on program models necessarily requires a rigorous mathematical formalism for reasoning about the system under construction, in other words – a logic. This explains the necessity of construction and investigation of logics that are oriented on program reasoning – *program logics*.

One cannot invent one universal program logic that would have all necessary properties. Therefore a hierarchy of logics has to be developed. Here we briefly introduce a hierarchy of logics, which reflects the hierarchy of CNPM. These logics are called *composition-nominative logics* (CNL).

Construction and investigation of the logic hierarchy is a complicated topic; some initial results were presented in [2-5]. In this paper we continue this work focusing on the satisfiability problem for the first-order CNL. Satisfiability is one of the essential problems in logic [6]. This problem is usually dual to the validity problem, and both problems are important from the point of view of formal methods.

To tackle the satisfiability problem in CNL we propose to use reduction techniques. We show that the satisfiability problem for the first-order CNL can be reduced to the same problem for the classical first-order logic. The reduction is not trivial, but it preserves the complexity of a formula in terms of formula length and reciprocal order of propositional connectives. Thus, existent methods and tools developed for solving the satisfiability problem for classical logics can be applied for solving instances of the satisfiability problem in composition-nominative logics.

The rest of the paper is structured as follows. In section 2 we give a motivational example of applying compositional-nominative approach for program formalization and logic construction. We define first-order CNL in section 3. In section 4 we consider the satisfiability problem for these logics and provide a reduction method for solving it. In section 5 conclusions are formulated.

Proofs are omitted here and will be provided in an extended version of the paper.

2 Constructing Program Logics: An Appetizer

Let us consider the example language EL which is used here to demonstrate how program logics can be constructed. EL is similar to such languages as WHILE [7], IMP [8], etc.

The grammar of the language is defined as follows:

$s ::= x := a \mid s_1 \,;\, s_2 \mid$ `if` b `then` s_1 `else` $s_2 \mid$ `while` b `do` $s \mid$ `begin` s `end`

$a ::= n \mid x \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid (a)$

$b ::= a_1 = a_2 \mid a_1 > a_2 \mid b_1 \vee b_2 \mid \neg b \mid (b)$,

where:

- n ranges over natural numbers $Nat = \{0, 1, 2, \ldots\}$,
- x ranges over variables (names) $V = \{M, N, \ldots\}$,
- a ranges over arithmetic expressions *Aexpr*,
- b ranges over Boolean expressions *Bexpr*,
- s ranges over statements (programs) *Stm*.

As an example consider an EL program GCD for calculating the greatest common divisor of two numbers:

`while` $\neg(M=N)$ `do` `if` $M>N$ `then` $M:=M-N$ `else` $N:=N-M$.

Starting from this example we construct step-by-step several program algebras:

- we first define semantics of GCD in the style of denotational semantics; as a result we obtain an algebra which includes n-ary mappings defined on natural numbers;
- then we define a simpler algebra by excluding n-ary mappings but introducing quasiary mappings instead;
- finally, we define a class of quasiary program algebras changing the set of natural numbers *Nat* to any set A. This class is a semantic base for program logics.

How can we represent program semantics? There are various approaches for formal semantics. Here we will follow denotational semantics [7, 8] representing semantic mapping as $[\![.]\!]$. On the abstract level semantics of GCD can be treated as a function. Given a state d in the form $[M \mapsto m, N \mapsto n]$, this function evaluates new values of M and N that are equal to the greatest common divisor of n and m. Having analyzed the structure of the program we see that it is constructed from symbols of binary subtraction function $-$, binary relations $=$ and $>$, unary logical connective $\neg$, variables M

and N, and from structuring constructs such as assignment, conditional statement, and loop. Note that subtraction function is partial on *Nat*. To emphasize mapping's partiality/totality we will use the sign $\xrightarrow{p}$ for partial mappings and the sign $\xrightarrow{t}$ for total mappings.

Symbols of arithmetic operations, relations, and Boolean connectives represent n-ary mappings defined on *Nat* or on the set $Bool=\{T, F\}$ of Boolean values.

For our language we define the following types of n-ary mappings:

$Fn^{n,Nat}=Nat^{\,n} \xrightarrow{p} Nat$, $Pr^{n,Nat}=Nat^{\,n} \xrightarrow{p} Bool$, $Pr^{\,n,Bool} = Bool^{\,n} \xrightarrow{p} Bool$, $n\geq 0$.

Using the same notation for language symbols and mappings they represent, we can write that $+, -, * : Fn^{\,2,Nat}$, $\vee: Pr^{\,2,Bool}$, $\neg : Pr^{\,1,Bool}$, $=, >: Pr^{\,2,Nat}$. (A natural number n can be considered as a null-ary function; but for simplicity's sake we skip it further formalization.)

To present formal semantics of arithmetic and Boolean expressions we first define the set of states $State = V \xrightarrow{p} Nat$ (denoted here as ${}^{V}Nat$). Arithmetic expressions denote functions (called *quasiary functions over Nat*) of the set $Fn^{V,Nat}=State \xrightarrow{p} Nat = {}^{V}Nat \xrightarrow{p} Nat$; thus, for $a\in Aexpr$ we have $[\![a]\!] \in Fn^{V,Nat}$. Boolean expressions denote predicates (called *quasiary predicates over Nat*) of the set $Pr^{V,Nat}= {}^{V}Nat \xrightarrow{p} Bool$; thus for $b\in Bexpr$ we have $[\![b]\!] \in Pr^{V,Nat}$.

Functions of the class $Fn^{V,Nat}$ and predicates of the class $Pr^{V,Nat}$ are also called *nominative functions* and *nominative predicates* respectively, since they are defined on states constructed with the help of naming (nominative) relation.

To represent semantics of variables in arithmetic expressions we will use a parametric denomination (denaming) function $'x$: $Fn^{V,Nat}$. Given a program state, the function $'x$ returns the value of the variable x in that state. For instance, denomination functions that yield values of names M and N are denoted as $'M$ and $'N$ respectively. These values may or may not be defined, so denomination functions are partial.

The semantics of structuring constructs is defined with the help of special operators over construct components' semantics. Such operators are called *compositions*. Thus, to define semantics of structured arithmetic and Boolean expressions, say of expressions $N–M$ and $N>M$ respectively, we will use superposition compositions which can be considered as a semantic explication of term substitution.

Superposition $S_F^n : Fn^{n,Nat} \times (Fn^{V,Nat})^n \xrightarrow{t} Fn^{V,Nat}$ of functions $g_1,\ldots, g_n$ into an n-ary function f^n is an operator such that $S_F^n\,(f^n, g_1,\ldots, g_n)(d) = f^n\,(g_1(d),\ldots,g_n(d))$ where d is a state. The same formula can be used for defining superposition $S_P^n : Pr^{n,Nat} \times (Fn^{V,Nat})^n \xrightarrow{t} Pr^{V,Nat}$ of functions $g_1,\ldots, g_n$ into n-ary predicate p^n, and for superposition $S_B^n : Pr^{n,Bool} \times (Pr^{V,Nat})^{\,n} \xrightarrow{t} Pr^{V,Nat}$ of predicates $p_1,\ldots, p_n$ into an n-ary Boolean function f_B^n. Thus, $[\![N–M]\!] = S_F^2\,(–, 'N, 'M)$, $[\![N>M]\!] = S_P^2\,(>, 'N, 'M)$, $[\![(N>M)\vee(M>N)]\!] = S_B^2\,(\vee,\ S_P^2\,(>, 'N, 'M),\ S_P^2\,(>, 'M, 'N))$.

Programs (statements) from *Stm* denote bi-quasiary functions of the class $Prg^{V,Nat}=State \xrightarrow{p} State = {}^{V}Nat \xrightarrow{p} {}^{V}Nat$. Such functions are also called *bi-nominative*.

Semantics of structured statements is defined by the following compositions with conventional meaning:

1. assignment composition AS^x: $Fn^{V,Nat} \xrightarrow{t} Prg^{V,Nat}$ (x is a parameter from V);
2. composition of sequential execution •: $Prg^{V,Nat} \times Prg^{V,Nat} \xrightarrow{t} Prg^{V,Nat}$;
3. conditional composition IF: $Pr^{V,Nat} \times Prg^{V,Nat} \times Prg^{V,Nat} \xrightarrow{t} Prg^{V,Nat}$;
4. loop composition WH: $Pr^{V,Nat} \times Prg^{V,Nat} \xrightarrow{t} Prg^{V,Nat}$.

For instance, $[\![M{:=}M{-}N\;]\!] = AS^M(S_F^2(-, 'M, 'N))$, $[\![M{:=}M{-}N;\; N{:=}M{-}N\;]\!] = AS^M(S_F^2(-, 'M, 'N)) \bullet AS^N(S_F^2(-, 'M, 'N))$. Note, that we define • by commuting arguments of conventional functional composition: $f \bullet g = g \circ f$.

Thus, we obtain the following program algebra with n-ary functions which has 7 carriers:

$$APn(V) = < Fn^{2,Nat}, Pr^{2,Bool}, Pr^{1,Bool}, Pr^{2,Nat}, Fn^{V,Nat}, Pr^{V,Nat}, Prg^{V,Nat};$$
$$+, -, *, \vee, \neg, =, >, S_F^2, S_P^2, S_B^2, S_B^1, 'x, AS^x, \bullet, IF, WH> .$$

We would like to emphasize the fact that semantics of EL programs (or EL expressions) can be represented as terms of this algebra. This simplifies investigations of EL programs because the constructed algebra completely specifies their semantics.

The term for EL program GCD is as follows:

$$WH(S_P^1(\neg, S_P^2(=,'M,'N)), IF(S_P^2(>,'M,'N), AS^M(S_F^2(-,'M,'N)), AS^N(S_F^2(-,'N,'M)))).$$

Note that this term and its sub-terms can denote partial mappings, as the subtraction and denomination functions can be undefined; also WH composition can be a source of undefinedness.

Having specified this algebra we can study properties of programs; this can be used in program reasoning. For example, we can prove associativity of sequential execution of statements: `begin` s_1 ; s_2 `end`; s_3 = s_1; `begin` s_2 ; s_3 `end` by proving in the algebra $APn(V)$ the corresponding property of composition of sequential execution: $f \bullet (g \bullet h) = (f \bullet g) \bullet h$. Another example is the following distributivity property:

`begin if` b `then` s_1 `else` s_2 `end`; s_3 =

= `if` b `then begin` s_1; s_3 `end else begin` s_2; s_3 `end`.

Its validity is based on the property of $APn(V)$ that $IF(p,f,g) \bullet h = IF(p, f \bullet h, g \bullet h)$. (Here $f, g, h \in Fn^{V,Nat}$, $p \in Pr^{V,Nat}$.)

Although these properties are valid for the algebra $APn(V)$ based on natural numbers, it is clear that we can consider other data types as *integer*, *real*, etc. So, one can ask a question: what EL program properties remain valid under type variations?

To understand this we should split the set of operation symbols of this algebra into two parts: logical symbols, which have relatively type-independent interpretations, and non-logical (or descriptive) symbols, which represent specifics of the carriers.

Trivial inspection of definitions shows that symbols +, –, $*$, =, > are descriptive symbols in the signature of $APn(V)$. They are defined over natural numbers, and may be considered constants in this algebra. Other symbols may be considered logical.

These considerations open the second phase of program algebra development. To make the algebra simpler we can exclude from it the classes of n-ary functions and predicates, thus concentrating on logical symbols that are interpreted as compositions over nominative carriers $Fn^{V,Nat}$, $Pr^{V,Nat}$, $Prg^{V,Nat}$. Still, n-ary functions and predicates can be represented in these classes of nominative mappings.

The idea of this representation, say for binary subtraction function, is the following: we represent a pair (n_1, n_2) as a state $[1 \mapsto n_1, 2 \mapsto n_2]$, where 1 and 2 should be treated as standard variables that represent the arguments of the binary function symbol. In order to avoid using standard names 1 and 2 and to obtain homogeneity of names we can introduce a parametric quasiary function $\boldsymbol{x-y}$ (printed in bold font) such that $\boldsymbol{x-y} = S_F^2(-, {}'x, {}'y)$; here x and y are parameters from V. In this case an expression $M-N$ will denote a quasiary function $\boldsymbol{M-N}$.

Therefore instead of binary functions we introduce parametric quasiary functions $\boldsymbol{x+y}$, $\boldsymbol{x-y}$, and $\boldsymbol{x*y}$; also instead of relations we introduce new parametric quasiary predicates $\boldsymbol{x=y}$ and $\boldsymbol{x>y}$ (with x and y as parameters).

This step permits to represent every n-ary function defined over Nat as a parametric quasiary function. But now, to represent the semantics of complex expressions we should introduce special superpositions $S_F^{v_1,\ldots,v_n}$ (or $S_F^{\bar{v}}$) and $S_P^{v_1,\ldots,v_n}$ (or $S_P^{\bar{v}}$), which are called *superpositions into quasiary function* and *predicate* respectively: $S_F^{v_1,\ldots,v_n}(f^q, g_1,\ldots, g_n)(d) = f^q(d\nabla[v_1 \mapsto g_1(d),\ldots,v_n \mapsto g_n(d)])$ and $S_P^{v_1,\ldots v_n}(p^q, g_1,\ldots, g_n)(d) = p^q(d\nabla[v_1 \mapsto g_1(d),\ldots,v_n \mapsto g_n(d)])$, where $f^q \in Fn^{V,Nat}$, $p^q \in Pr^{V,Nat}$.

Intuitive meaning of these formulas is that we change in d the values of names $v_1,\ldots, v_n$ to $g_1(d),\ldots,g_n(d)$ respectively (partiality should be taken into account). Thus, semantics, say of the expression $M+(N-K)$, can be represented as $S_F^N(\boldsymbol{M+N}, \boldsymbol{N-K})$.

Now let us consider logical symbols $\vee$: $Bool^2 \xrightarrow{t} Bool$ and $\neg$: $Bool \xrightarrow{t} Bool$. We cannot directly represent them as quasiary predicates, therefore we advocate another approach. We will treat them as binary compositions over quasiary predicates (denoted by the same signs) $\vee$: $Pr^{V,Nat} \times Pr^{V,Nat} \xrightarrow{t} Pr^{V,Nat}$ and $\neg$: $Pr^{V,Nat} \xrightarrow{t} Pr^{V,Nat}$. Such representations also provide better possibilities to work with partial predicates. For example, consider a Boolean expression $(M-N)>M \vee (M>L)$. Its semantics in $APn(V)$ is represented by the term $S_B^2(\vee, S_P^2(>, S_F^2(-, {}'M, {}'N), {}'M), S_P^2(>, {}'M, {}'L)))$. But superposition into an n-ary mapping is strict: when one argument is not defined then the result is also undefined. This restricts usage of partial predicates. For example, for Kleene's strong disjunction [9] it is allowed that one argument may be undefined if the other one is evaluated to true. When we represent connectives as compositions, we avoid the above considered difficulties.

Thus, we can now consider a simpler algebra – the program algebra of quasiary predicates with constants:

$APQC(V) = < Fn^{V,Nat}, Pr^{V,Nat}, Prg^{V,Nat};$

$$\boldsymbol{x-y}, \boldsymbol{x+y}, \boldsymbol{x*y}; \boldsymbol{x=y}, \boldsymbol{x>y}; \vee, \neg, S_F^{\bar{v}}, S_P^{\bar{v}}, {}'x, AS^x, \bullet, IF, WH>.$$

In this algebra we have 5 parametric descriptive symbols considered as algebra's constants, and 9 logical symbols. Note, that all logical symbols are treated as compositions, possibly null-ary compositions (as in the case of denomination functions).

In this algebra semantics of GCD program is represented by the following term:

$$WH(\neg\, (\boldsymbol{M{=}N}), IF(\boldsymbol{M{>}N}, AS^{M}(\boldsymbol{M{-}N}), AS^{N}(\boldsymbol{N{-}M}))).$$

The next step of constructing more "logical" algebras consists of eliminating from $APQC(V)$ descriptive symbols $\boldsymbol{x{-}y}$, $\boldsymbol{x{+}y}$, $\boldsymbol{x{*}y}$, $\boldsymbol{x{=}y}$, $\boldsymbol{x{>}y}$ with fixed interpretations. We obtain a program algebra of quasiary predicates $APQ(V)$ without constants:

$$APQ(V) = < Fn^{V,Nat}, Pr^{V,Nat}, Prg^{V,Nat};\ \vee, \neg,\ S_F^{\bar{v}},\ S_P^{\bar{v}}, 'x,\ AS^{x}, \bullet, IF, WH >.$$

As to descriptive symbols, we can instead consider two sets Fs and Ps of function and predicate symbols that do not have predefined interpretations, and consequently, can denote any quasiary function or predicate.

As we are interested in general laws of reasoning about programs, we should make the next step and define compositions for every set A of basic (atomic) values. In this case we obtain the following program algebra of quasiary predicates:

$$APQ(V, A) = < Fn^{V,A}, Pr^{V,A}, Prg^{V,A};\ \vee, \neg,\ S_F^{\bar{v}},\ S_P^{\bar{v}}, 'x,\ AS^{x}, \bullet, IF, WH >.$$

Symbols from Fs and Ps are used to construct terms of this algebra. Properties of such terms are general properties because they should be valid under any interpretations of function and predicate symbols.

It means that we have constructed a class of program algebras (for various A), representing program semantics for languages with different domains. Such algebras may be called *general program models*; they form the semantic base for program logics.

For example, we can consider equational program logics by defining formulas of these logics as formal equalities of the form $t_1=t_2$, where t_1 and t_2 are terms of the type $Prg^{V,A}$. Such logics define equivalent transformations of programs.

Another conventional program logic is Floyd–Hoare logic, which is based on assertions of the form $\{b_1\}s\{b_2\}$. Semantics of such assertions can be presented by Floyd–Hoare composition FH: $Pr^{V,A} \times Prg^{V,A} \times Pr^{V,A} \xrightarrow{t} Pr^{V,A}$. We define this composition under assumption that predicates and functions can be partial. Then

$$FH(p, prg, q)(d) = \begin{cases} T, & \text{if } q(prg(d)) \downarrow = T \text{ or } p(d) \downarrow = F, \\ F, & \text{if } p(d) \downarrow = T \text{ and } q(prg(d)) \downarrow = F, \\ & \text{undefined in other cases.} \end{cases}$$

Here we write $p(d)\downarrow$ if a predicate p is defined on data d, $p(d)\downarrow = r$ if a predicate p is defined on data d with a value r, $p(d)\uparrow$ if a predicate p is undefined on d. For nominative data representation we use the form $d = [v_i \mapsto a_i \mid i \in I]$. Extending the algebra $APQ(V,A)$ with FH composition, a parametric composition of existential quantification $\exists x$: $Pr^{V,A} \xrightarrow{t} Pr^{V,A}$, and a composition of equality = we obtain the algebra

$$APFH(V, A) = < Fn^{V,A}, Pr^{V,A}, Prg^{V,A};\ \vee, \neg,\ S_F^{\bar{v}},\ S_P^{\bar{v}}, 'x, \exists x, =, AS^{x}, \bullet, IF, WH, FH >.$$

The class of such algebras forms the semantic base for quite powerful Floyd–Hoare-like logics of quasiary mappings. It is important to admit that by restricting this algebra on one and two carriers we obtain respectively the following two algebras:

– propositional algebra $AP(V,A) = <Pr^{V,A};\ \vee, \neg>$;
– first-order algebra $AFO(V, A) = <Pr^{V,A}, Fn^{V,A}; \vee, \neg,\ S_F^{\bar{v}},\ S_P^{\bar{v}}, 'x, \exists x, =>$.

Propositional compositions are defined as follows ($p, q \in Pr^{V,A}$, $d \in {}^{V}A$):

$$(p \vee q)(d) = \begin{cases} T, \text{ if } p(d)\downarrow= T \text{ or } q(d)\downarrow= T, \\ F, \text{ if } p(d)\downarrow= F \text{ and } q(d)\downarrow= F, \\ \text{undefined in other cases.} \end{cases} \qquad (\neg p)(d) = \begin{cases} T, \text{ if } p(d)\downarrow= F, \\ F, \text{if } p(d)\downarrow= T, \\ \text{undefined if } p(d)\uparrow. \end{cases}$$

Unary parametric composition of existential quantification $\exists x$ with the parameter $x \in V$ is defined by the following formula ($p \in Pr^{V,A}$, $d \in {}^{V}A$):

$$(\exists x\ p)(d) = \begin{cases} T, \quad \text{if } b \in A \text{ exists}: \ p(d\nabla x \mapsto b)\downarrow= T, \\ F, \quad p(d\nabla x \mapsto a)\downarrow= F \ \text{ for each } a \in A, \\ \qquad \text{undefined in other cases.} \end{cases}$$

The ∇ operation is defined as follows: if d_1 and d_2 are nominative data then $d = d_1\nabla d_2$ consists of all named pairs of d_2 and only those pairs of d_1, whose names are not defined in d_2. A shorter form for $d\nabla[x \mapsto a]$ is $d\nabla x \mapsto a$.

Null-ary parametric denomination composition with the parameter $x \in V$ is defined by the following formula ($d \in {}^{V}A$): $'x(d) = d(x)$.

Binary equality composition = is defined as follows ($f, g \in Fn^{V,A}$, $d \in {}^{V}A$):

$$(f{=}g)(d) = \begin{cases} T, \text{ if } f(d)\downarrow,\ g(d)\downarrow \text{ and } f(d) = g(d), \\ T, \text{ if } f(d)\uparrow \text{ and } g(d)\uparrow, \\ F \text{ otherwise.} \end{cases}$$

Based on classes of such algebras various propositional, first-order, and program CNL can be defined. We can go even further and define modal and temporal CNL [4].

Summing up, we would like to say that semantics of programs can be presented by terms of program algebras with compositions as operations of this algebra; program, functions, and predicates are defined on nominative sets (nominative data); we define program logics directly on program algebras by extending their signatures with special “logical” compositions.

In the rest of the paper we consider first-order composition-nominative logics. We will study the satisfiability problem in these logics and apply reduction methods for solving this problem.

3 Formal Definitions of First-Order Composition-Nominative Logics

We use the semantic-syntactic scheme of logic definition [4, 10]. It means that we first define a class of algebras, which forms logic’s semantics (this was done in the previous section), then we define a language of the logic, and at last, interpretations of functions and predicate symbols are described. Recall that for simplicity’s sake we use the same notations for compositions (as operations in the algebra) and their symbols.

A tuple $\Sigma = (V, \{\vee, \neg, S_P^{\bar{v}}, S_F^{\bar{v}}, 'x, \exists x, =\}, Fs, Ps)$ is called *a signature* of the first-order composition-nominative logic. Taking into consideration that a set of composition symbols is determined by the set V, we will use for a signature a simplified notation (V, Fs, Ps). To define the language $L(\Sigma)$ we define the class of its terms $Tr(V, Fs, Ps)$ and the class of its formulas $Fr(V, Fs, Ps)$, which are constructed over terms.

Inductive definition of $Tr(V, Fs, Ps)$ is as follows:

– if $F \in Fs$ then $F \in Tr(V, Fs, Ps)$;
– if $x \in V$ then $'x \in Tr(V, Fs, Ps)$;
– if $\bar{v} = (v_1,...,v_n)$ is a list of distinct variables ($n \geq 0$), $t, t_1,\dots,t_n \in Tr(V, Fs, Ps)$ then $S_F^{\bar{v}}(t, t_1,...,t_n) \in Tr(V, Fs, Ps)$.

Inductive definition of $Fr(V, Fs, Ps)$ is as follows:

– if $P \in Ps$ then $P \in Fr(V, Fs, Ps)$;
– if $\Phi, \Psi \in Fr(V, Fs, Ps)$ then $(\Phi \vee \Psi) \in Fr(V, Fs, Ps)$ and $\neg\Phi \in Fr(V, Fs, Ps)$;
– if $\Phi \in Fr(V, Fs, Ps)$, $\bar{v} = (v_1,...,v_n)$ is a list of distinct variables ($n \geq 0$), $t_1,\dots, t_n \in Tr(V, Fs, Ps)$ then $S_P^{\bar{v}}(\Phi, t_1,...,t_n) \in Fr(V, Fs, Ps)$;
– if $x \in V$, $\Phi \in Fr(V, Fs, Ps)$ then $\exists x\Phi \in Fr(V, Fs, Ps)$;
– if $t_1, t_2 \in Tr(V, Fs, Ps)$ then $t_1 = t_2 \in Fr(V, Fs, Ps)$.

To finish the definition of the logic we need to specify interpretation mappings for function and predicate symbols. This is done with mappings $I^{Fs} : Fs \xrightarrow{t} Fn^{V,A}$ and $I^{Ps} : Ps \xrightarrow{t} Pr^{V,A}$ respectively. Then we can compositionally construct the interpretation mappings for terms and formulas. A triple ($AFO(V, A)$, I^{Fs}, I^{Ps}) is called *a model for* $L(\Sigma)$. A model is determined by a tuple $J^{Fs,Ps} = (V, A, I^{Fs}, I^{Ps})$ called *an interpretation*. In the simplified form interpretations will be denoted as J. For an interpretation J and a formula Φ the meaning of Φ in J is denoted Φ_J. Interpretation mechanism is traditional.

As there is no distributivity of existential quantifier with superposition compositions [4] we need an infinite set of unessential names in order to be able to carry out equivalent transformations of formulas. We assume that a set U of unessential variables is an infinite subset of V ($U \subseteq V$). Informally speaking, this restricts the class of possible interpretations but does not affect the satisfiability problem [10].

4 Reduction of Satisfiability Problem for the First-Order CNL

In this paper we present a result that the satisfiability problem in the first-order CNL can be reduced to the satisfiablility problem in the classical first-order predicate logic. This result can be seen as a generalization of the reduction of the satisfiability problem for CNL of quantifier-equational level [10].

A formula Φ is called *satisfiable in an interpretation* J if there is $d \in {}^{V}\!A$ such that $\Phi_J(d)\downarrow = T$. We shall denote this by $J \mathrel{|\!\!\approx} \Phi$. A formula Φ is called *satisfiable* if there exists an interpretation J in which Φ is satisfiable. We shall denote this $\mathrel{|\!\!\approx} \Phi$. We call formulas Φ and Ψ *equisatisfiable* if they are either both satisfiable or both not satisfiable (i.e., unsatisfiable).

The problem is to check whether $\models\approx\Phi$ holds given an arbitrary formula $\Phi\in Fr(V, Fs, Ps)$. Our main aim is to transform the CNL formula Φ to an equisatisfiable formula Φ_{CL} of the classical first-order logic with equality so that we can use existent methods for solving this problem developed for classical logics.

We transform the formula Φ to a special normal form, which is translated to its classical counterpart Φ_{CL} afterwards.

A formula Φ is said to be in *unified superposition normal form* (USNF) if the following requirements are met:

- for every sub-formula of the form $S_P^{\bar{v}}(\Psi,\bar{t})$ we have that $\Psi\in Ps$;
- for every sub-formula of the form $S_F^{\bar{v}}(r,\bar{t})$ we have that $r\in Fs$;
- all instances of superposition compositions have the same list of variables $\bar{w}$;
- for every quantifier $\exists y$ occurring in Φ we have that y should occur in $\bar{w}$ (see the previous rule).

Consider equivalent transformation rules T1-T12 of the form $\Phi_l \mapsto \Phi_r$, $\Phi_l, \Phi_r \in Fr(V, Fs, Ps)$. These rules are equivalent transformations in CNL [4].

T1) $S_P^{\bar{v}}(\Phi\vee\Psi,\bar{t}) \mapsto S_P^{\bar{v}}(\Phi,\bar{t}) \vee S_P^{\bar{v}}(\Psi,\bar{t})$.

T2) $S_P^{\bar{v}}(\neg\Phi,\bar{t}) \mapsto \neg S_P^{\bar{v}}(\Phi,\bar{t})$.

T3) $S_P^{\bar{v}}(\exists x\Phi,\bar{t}) \mapsto \exists u\ S_P^{\bar{v}}(S_P^{x}(\Phi,'u),\bar{t})$, u is an unessential variable that does not occur in $S_P^{\bar{v}}(\exists x\Phi,\bar{t})$, $u\in U$.

T4) $S_P^{\bar{u},\bar{x}}(S_P^{\bar{x},\bar{v}}(\Phi,\bar{r},\bar{s}),\bar{t},\bar{w}) \mapsto S_P^{\bar{u},\bar{x},\bar{v}}(\Phi,\bar{t}, S_F^{\bar{u},\bar{x}}(r_1,\bar{t},\bar{w}),\dots, S_F^{\bar{u},\bar{x}}(r_k,\bar{t},\bar{w}), S_F^{\bar{u},\bar{x}}(s_1,\bar{t},\bar{w}),\dots, S_F^{\bar{u},\bar{x}}(s_m,\bar{t},\bar{w}))$, here (and in T5) $\bar{u}=u_1,\dots,u_n$; $\bar{t}=t_1,\dots,t_n$; $\bar{x}=x_1,\dots,x_k$; $\bar{r}=r_1,\dots,r_k$; $\bar{w}=w_1,\dots,w_k$; $\bar{v}=v_1,\dots,v_m$; $\bar{s}=s_1,\dots,s_m$, $u_i\neq v_j, i=1,\dots,n, j=1,\dots,m$.

T5) $S_F^{\bar{u},\bar{x}}(S_F^{\bar{x},\bar{v}}(t,\bar{r},\bar{s}),\bar{t},\bar{w}) \mapsto S_F^{\bar{u},\bar{x},\bar{v}}(t,\bar{t}, S_F^{\bar{u},\bar{x}}(r_1,\bar{t},\bar{w}),\dots, S_F^{\bar{u},\bar{x}}(r_k,\bar{t},\bar{w}), S_F^{\bar{u},\bar{x}}(s_1,\bar{t},\bar{w}),\dots, S_F^{\bar{u},\bar{x}}(s_m,\bar{t},\bar{w}))$.

T6) $S_P^{\bar{v}}(r=s,\bar{t}) \mapsto S_F^{\bar{v}}(r,\bar{t}) = S_F^{\bar{v}}(s,\bar{t})$.

T7) $S_P^{\bar{v}}(\Phi,\bar{t}) \mapsto S_P^{x,\bar{v}}(\Phi,'x,\bar{t})$, x does not occur in $\bar{v}$. In particular $\Phi\mapsto S_P^{x}(\Phi,'x)$.

T8) $S_F^{\bar{v}}(t,\bar{t}) \mapsto S_F^{x,\bar{v}}(t,'x,\bar{t})$, x does not occur in $\bar{v}$. In particular, $t\mapsto S_F^{x}(t,'x)$.

T9) $S_P^{\bar{u},x,\bar{v}}(\Phi,\bar{q},r,\bar{s}) \mapsto S_P^{x,\bar{u},\bar{v}}(\Phi,r,\bar{q},\bar{s})$.

T10) $S_F^{\bar{u},x,\bar{v}}(t,\bar{q},r,\bar{s}) \mapsto S_F^{x,\bar{u},\bar{v}}(t,r,\bar{q},\bar{s})$.

T11) $S_F^{x,\bar{v}}('x,t,\bar{r}) \mapsto t$.

T12) $S_F^{\bar{v}}('x,\bar{r}) \mapsto 'x$, x does not occur in $\bar{v}$.

Rules T3 and T7 permit to assume w.l.o.g. that all quantified variables in initial formula are different.

Given an arbitrary formula $\Phi \in Fr(V, Fs, Ps)$ we can construct (non-deterministically) its unified superposition normal form $usnf[\Phi]$ by applying rules T1-T12. This transformation is satisfiability-preserving.

In order to reduce the satisfiability problem in the first-order CNL to the satisfiability problem for classical logics we consider algebras defined over extended data set $A_\varepsilon^V = V \xrightarrow{t} A \cup \{\varepsilon\}$. Informally, the additional value ε represents undefined components of nominative data. So, by changing data classes from ${}^V A$ to A_ε^V we make our algebras closer to their classical counterparts and simplify required proofs.

We formalize the syntactical reduction *clf* of terms and formulas in unified superposition normal form to formulas of classical logic inductively as follows:

1. $clf['x] \mapsto x.$
2. $clf[S_F^{w_1,\dots,w_n}(F, t_1,\dots,t_n)] \mapsto F(clf[t_1],\dots,clf[t_n])$, $F \in Fs.$
3. $clf[(\Phi_1 \vee \Phi_2)] \mapsto (\, clf[\Phi_1] \vee clf[\Phi_2])$.
4. $clf[\neg\Phi] \mapsto \neg clf[\Phi]$.
5. $clf[S_P^{w_1,\dots,w_n}(P, t_1,\dots,t_n)] \mapsto P(clf[t_1],\dots,clf[t_n]),\ n \geq 0$.
6. $clf[\exists x\Phi] \mapsto \exists x(x \neq e \;\&\; clf[\Phi])$, $e \in U$, e is a predefined variable.
7. $clf[t_1 = t_2] \mapsto clf[t_1] = clf[t_2]$.

Note that all applications of the 6-th rule introduce the same variable e which is a predefined variable from U in the sense that it does not occur in USNF. In interpretations the value of this variable is treated as ε.

This reduction transforms the formula to a formula of classical logic preserving its satisfiability.

Theorem. Let $\Phi \in Fr(V, Fs, Ps)$. Then $\models\!\approx \Phi$ if and only if $clf[usnf[\Phi]]$ is satisfiable in the classical first-order predicate logic.

The theorem states the reduction of the satisfiability problem in the first-order composition-nominative logic to the satisfiability problem in the classical first-order predicate logic.

5 Conclusions

Rigorous approaches for software systems development can be based on different logics. In this paper we have discussed the composition-nominative approach, its motivation, main principles, and logics evolved in this approach. The overview of composition-nominative logics (CNL) is given and their application in formal approaches for software development is discussed. CNL are algebra-based logics of partial predicates constructed in a semantic-syntactic style on the methodological basis, which is common with programming. They can be considered as generalizations of traditional logics on classes of partial predicates that do not have fixed arity. In the paper we deal with the first-order CNL. This logic is one of the most important in the CNL hierarchy. As a main result we have shown that the satisfiability problem for the first-order CNL can be reduced to the satisfiability problem for classical first-order logic with equality. Therefore, existent state-of-the-art methods and techniques for checking satisfiability in classical logics can also be applied to CNL.

Future work on the topic will include investigation of satisfiability problem for richer CNL over hierarchic nominative data. Hierarchic data permit to represent such complex structures as lists, stacks, arrays, etc.; thus, such logics will be closer to program models with more rich data types. Another direction is related with identification of classes of formulas in various types of CNL for which the satisfiability problem can be solved efficiently. In particular, this concerns specialized theories, where some predicates have specific interpretations and certain axioms shall hold for such interpretations. This is often referred to as satisfiability modulo theory (SMT) problem [11]. Also, prototypes of software systems for satisfiability checking in CNL should be developed.

Acknowledgements. The authors would like to thank Andrii Kryvolap and Dr. Wolfgang Schreiner for helpful comments on earlier drafts of the paper.

References

1. George, C., Haxthausen, A.E., Hughes, S., et al.: The RAISE Development Method. Prentice Hall, London (1995)
2. Nikitchenko, N.S.: A Composition Nominative Approach to Program Semantics. Technical Report IT–TR 1998-020, Technical University of Denmark (1998)
3. Basarab, I.A., Gubsky, B.V., Nikitchenko, N.S., Red'ko, V.N.: Composition Models of Databases. In: Eder, J., Kalinichenko, L.A. (eds.) East-West Database Workshop (Workshops in Computing Series), pp. 221–231. Springer, London (1995)
4. Nikitchenko, M.S., Shkilnyak, S.S.: Mathematical logic and theory of algorithms. Publishing house of Taras Shevchenko National University of Kyiv (2008) (in Ukrainian)
5. Nikitchenko, M.S.: Composition-nominative aspects of address programming. Kibernetika I Sistemnyi Analiz 6, 24–35 (2009) (in Russian)
6. Mendelson, E.: Introduction to Mathematical Logic, 4th edn. Chapman & Hall, London (1997)
7. Nielson, H.R., Nielson, F.: Semantics with Applications: A Formal Introduction. John Wiley & Sons Inc. (1992)
8. Winskel, G.: The Formal Semantics of Programming Languages. MIT Press, Cambridge (1993)
9. Kleene, S.C.: Introduction to Metamathematics. Van Nostrand, New York (1952)
10. Nikitchenko, M.S., Tymofieiev, V.G.: Satisfiability Problem in Composition-Nominative Logics of Quantifier-Equational Level. In: Proc. 8th Int. Conf. ICTERI 2012, Kherson, Ukraine, June 6-10 (2012), CEUR-WS.org/Vol-848, `http://ceur-ws.org/Vol-848/ICTERI-2012-CEUR-WS-paper-38-p-56-70.pdf`
11. Barrett, C., Sebastiani, R., Seshia, S.A., Tinelli, C.: Satisfiability Modulo Theories. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability. IOS Press (2009)

Adaptable Enterprise Information Systems Development Using Advanced Active Data Dictionary Framework

Maxim Davidovsky[1], Gennadiy Dobrovolsky[1], Olga Todoriko[2], and Vladimir Davidovsky[1]

[1] Zaporozhye National University, Center of Information Technologies, Zhukovskogo Str. 66, 69600 Zaporozhye, Ukraine
[2] Zaporozhye National University, Department of Information Technologies, Zhukovskogo Str. 66, 69600 Zaporozhye, Ukraine
m.davidovsky@gmail.com, o-sun@rambler.ru, {gen,dvm}@znu.edu.ua

Abstract. Today management activities pose new challenges for corporate information systems. The rate of changing and dynamics is increasing in the course of time as well as complexity of tasks to be solved that requires adequate reactions from agent of management. In such conditions information systems have to have a set of characteristics such as versatility, flexibility, and scalability to effectively address the goals. The paper presents our ongoing work at a generic framework for developing flexible enterprise information systems based on the concept of the Advanced Active Data Dictionary (AADD). The framework allows construction of flexible enterprise applications for efficient solutions in various domains and broad range of management activities.

Keywords: Corporate system, active data dictionary, software architecture, application development, framework.

Key Terms. SoftwareEngineeringProcess, ModelBasedSoftwareDevelopmentMethodology, Development, Management.

1 Introduction

The progress of modern society is accompanied by significant acceleration of processes that leads to rapid changes in informational background and requires more agile and adequate responses to the changes. In different management activities such as managerial decision-making in business, control and regulation of social processes or support for political decision-making efficient agents of management have to act faster and more adequate according to the situation. Such circumstances bring to the forefront the requirements of consistency and complexity of developing automation systems.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 152–161, 2013.

Another important issue is possible changes in scope and scale of managed objects that demands mobility and scalability from information systems (IS). The higher level of potential integrability laid into a developed system, the longer life circle the system will have. It allows shifting greater part of routine tasks concerning data acquisition, systematization, transmission and analysis to the automated systems. Moreover, increasing performance of computer systems and progress in information representation and reconciliation methods enlarge the scope of tasks that can be automated and enhance effectiveness of the solutions. Heterogeneity in ISs integration is one more challenge to be solved. State-of-the-art ISs have to interact with independent external services. Unfortunately, such services could become unavailable due to network errors. Moreover, their APIs can be changed independently that leads to contradictions with current ISs' structure and data management principles. Thus, failures of systems with rigid structure occur very often. Similar problems concern the end-user application. It should be tolerant to accidental environment variation and flexible enough to be adapted, deployed and customized for a wide class of applications in various areas of management activities – in business, in government, in educational institutions, etc. Dynamics and variability of subject domains and customer needs lead to the situation where logic, data structures and visual representation are very dynamic and can change not only during the process of designing and developing of ISs, but during their daily exploitation.

Foregoing requirements can be fulfilled by using appropriate theoretical framework and implementing it into appropriate toolset. To the best of our knowledge the most appropriate frameworks are Active Data Dictionary (ADD) [1], Event-Driven Architecture (EDA) [2] and Enterprise Service Bus (ESB) [3].

In the paper we present the concept of Advanced Active Data Dictionary (AADD) and provide preliminary designing details. We propose an exokernel-based software architecture laying the foundation for an implementation of the framework for flexible corporate systems (CS). In section 2 we analyze features of efficient and flexible CS and articulate a set of requirements for implementing software. Being based on the requirements we state objectives of our work. In section 3 we analyze related theoretical frameworks and overview a number of solutions aimed at resolution of similar challenges. The rest of the paper is organized as follows. Section 4 reports about our contribution and section 5 concludes the paper with the analysis of intermediate results and plans for future work.

2 Objectives

For implementing the functionality briefly discussed in the previous section, it must be developed a software system functional basis that allows building of automatic management system (AMS) class applications by modeling (of subject domain) and configuring of an application. The AADD-based basis is a set of dictionaries and system tables (the tables used only by AADD system and not accessible for application) that allows describing the internal structure of an application and external objects interacting with it. For example, the structure and relationships between persistent data, the composition of application components and their relationships, etc. External objects are groups of users and individual users, external systems that interact with the application, operating platforms, data warehouses and so on. In addition, the AADD functional basis must include minimum necessary and sufficient set of functionality that provides control of the AADD.

A system based on the basis can be distributed over local and global network. Obviously, some parts of the system can be independent, unreliable or frequently changed. Despite of this the system must work reliably and smoothly. Hence, there must be no deadlocks and long waiting threads in the system. When a certain service becomes unavailable the system must work on. The AADD-based application should be of a three-level structure, which includes the client side (like a thin client and the client component of the application), the server side (implements the core functionality) and an interface to the data store(s). AADD components should be as independent as possible from the (OS) platform and the specific implementation of a data store. The system must be highly configurable. When external service API is changed administrator can adapt the system to new interface by updating the configuration. Existing program code rarely needs to be changed. This feature implies that activation/deactivation/changes of modules are performed by setting the configuration. It requires universal interface between moduless and transparent network communication. The uniform communication provides a basis for combination of simple modules into the complex constructs and therefore allows complex event processing.

Finally, system exploitation should require minimal programming skills (from an administrator), and standard computer skills (from a user). The system should allow carrying out system modification easily and with minimal involvement of a programmer both at the design stage and during the exploitation.

3 Related Work

The solution of the aforementioned tasks can be accomplished via implementation of a number of theoretical concepts. Below, we briefly describe some of the concepts and theoretical frameworks that allow creating multipartite applications with a wide-range scaling and satisfy the requirements of mobility. We compare core features of the alternative frameworks with the proposed in the paper and analyze their applicability. Further we present few implemented systems based on similar frameworks. A detailed analysis of all possible concepts and implementations lies beyond the scope of the article. Here we provide a short reference list and outline the scope of potentially useful approaches.

3.1 Theoretical Frameworks

Firstly we sketch the key concepts used in our framework.

Active Data Dictionary. The concept of ADD is known and well founded and is used widely enough in enterprise applications having substantial database backend (for instance, see [1] or [4]). In database development ADD represents a metadata level that provides means for data access with taking into account relationship between data. Using of ADD allows configuring of an information system on upper level by setting the metadata. In our project we refine the concept of ADD and develop a concept of the Advanced Active Data Dictionary (AADD). On the one hand, AADD contains not only information about data structure, types, properties and relationships but also information about the elements of interface, data visualization methods, document templates, characteristics and privileges of user groups and roles of individual users. It provides means to create and modify data structures, information system structure, implemented business processes simply by entering respective information into the data dictionary and an application can be configured with no (or little) time spent on programming. On the other hand, it should be developed an automated system for AADD management, which in turn should contain a set of features such as data visualization (in tabular form), menu construction, report building, data entry windows, etc. These features are characteristic for any management system. Thus, developing the core of a software application implementing the functions of AADD management thereby provide a functionally complete and thorough toolset for designing any application requiring automatic management system functionality (such as ERP[1], MRP[2], ACS[3] or CRM[4] systems).

Event-Driven Architecture (EDA). EDA provides minimal coupling of system parts, publisher-subscriber relations, and asynchronous communication. In EDA publisher doesn't know its subscribers as well as subscriber doesn't know its publisher. Both deal only with data transmission channel. Publisher-subscriber communication provides many-to-many messaging (i.e. published message is potentially available to all system parts which are authorized). The sent message does not require instant reaction. In addition, EDA is a simple way of Complex Event Processing (CEP) i.e. detection and processing of simple event combinations [5, 6].

Enterprise Service Bus (ESB). ESB provides service addresses, adapters and message routing. Service adapter is responsible for converting custom service data structures into universal messages. Each service connected to ESB has a unique address and the messages are routed to appropriate receivers. Also ESB contains route configuring tool.

Exokernel. The concept of exokernel comes from operation system development [7]. In our framework the exokernel provides only inter-process communication, safe allocating and freeing of resources. Such solution enables varied modules to cooperate through the mediation of exokernel. System parts can be easily added and removed smoothly without rebooting.

There are some alternative approaches that are not used in our framework, but also allow building ISs that meet the requirements articulated in the second section.

Technology of Active Objects (TAO). TAO has been emerged as a result of the integration of object-oriented paradigm (OOP) with the apparatus of underdetermined models (N-models) and the method of underdetermined calculation (N-Computing) [8] which was implemented in a number of experimental systems (e.g. [9]).

---

[1] Abbr. for Enterprise Resource Planning.

[2] Abbr. for Manufacturing Resource Planning.

[3] Abbr. for Automatic Control System.

[4] Abbr. for Customer relationship management.

The main components of domain model in the systems are objects. Objects can represent any domain entities and have all basic features of OOP objects. However, unlike OOP objects active objects contain no algorithmic programs but computational model that is presented declaratively in the form of constraints over object slot values as well as external variables. An active object can independently react to the behavior of other objects that are "visible" for it and signals coming from external sources (e.g. mouse, keyboard, other programs, etc.), and change its state. Interaction between active objects is asynchronous and based on the mechanism of control over the data. However, existing TAO solutions do not have appropriate set of tools that is enough for communication with the outness. In particular, there is no output of information generated by an active object into a generic external receiver (e.g. a file, another program, the physical device, etc.) that allows active objects not only react to external events, but also to affect on the environment. Interaction with a user is also not implemented.

In our solution objects have all basic properties in OOP sense. Moreover:

1. an object has internal memory, a state may adopt different signals (messages) from outside and send signals (messages) to external environment;
2. an object can contain (imperative) algorithmic programs as well as computational model that is presented as a set of declarations describing a data flow within the object. Declaration may take the form of function calls, connecting the values of object slots. (Here we assume a slot as a number of incoming messages of a certain type with specific syntax and semantics);
3. an active object can independently respond to the behavior of other "visible" objects and signals coming from external sources;
4. implement interaction with a user;
5. the system doesn't require any special skills from an administrator and user.

Reactive Programming [10]. The main focuses of reactive programming paradigm are data flow and dissemination of changes. This means that it should be possible to express the static or dynamic flow of data via a programming language, and execution model automatically distributes the changes along the data flow. A typical example of the state-of-the-art reactive programming is spreadsheet[5]. Reactive programming has been proposed as a way to simplify the creation of interactive graphical user interfaces, animations and real-time systems, but became more widespread as a common programming paradigm.

In our solution programming of a module running in the synchronous dataflow control: the module knows what kind of message it can send and receive, but does not know actual senders or recipients. Interaction between the modules is defined by system configuration. This is a slight deviation from the synchronous data-flow. The system allows activating and deactivating modules dynamically, but in practice this is rarely needed and real distinctions from synchronous dataflow are minimal.

Actor model (AM). AM is a mathematical model of parallel computing, which deals with the concept of "actor" as a generic entity in parallel numerical computation.

---

[5] A Brief History of Spreadsheets – `http://dssresources.com/history/sshistory.html`

As response to the messages it receives, an actor can make local decisions, create new actors, send messages, and establish how to respond to subsequent messages. AM appeared in 1973 [11] and was used as a basis for understanding the calculation of processes and the theoretical basis for a number of practical implementations of parallel systems. A good example of the actor model implementation is Akka[6]. It uses the model to increase the level of abstraction and provide a platform for building scalable parallel applications. Actors also provide an abstraction for distribution of programs across multiple processors. However, it should be mentioned that the Akka framework only provides means for communication and monitoring. Construction of ready-to-use system is assigned to skilled programmers who need to create addressing and routing systems, define actors types and functions, specify the relations between actors and so on. Additionally, for the implementation of the abovementioned tasks, it is needed to specify the structure of modules, activation rules, interactions and turnoffs.

3.2 Related Implementations

To the best of our knowledge there are only 2 ready-to-use systems that are directly comparable to our framework – Adempiere[7] and Compiere[8]. It should be mentioned that both systems have very similar internal model but only use different means for implementation. Actually Adempiere is a free implementation of Compiere, rewritten without the use of commercial libraries and APIs that are used in the maternal system.

Compiere and Adempiere are fully integrated software solutions for automating (geographically distributed) business. The systems are recommended for medium-size enterprises. Similarly to our framework they utilize the concept of a central active data dictionary (ADD). ADD contains:

- data structure definitions and how data is displayed in the forms, reports, and help wizards
- the order of relative fields
- workflows between forms
- security and access rules

The systems are implemented using J2EE[9] stack of technologies and are oriented to the use of the Oracle[10] database management system (DBMS) (but several other DBMSs are also supported). Basic functionality of the systems is implemented for ERP on a client machine (that allows for a higher level of security, compared with a thin web-client). For CRM and e-commerce applications the functionality is implemented on a server side. All implemented business processes are constructed with a built-in Business Process Management System (BPMS) (in contrast, in traditional ERP and CRM systems business logic is implemented in code). This

[6] `http://akka.io/`
[7] `http://www.adempiere.ru/`
[8] `http://www.compiere.com/`
[9] `http://www.oracle.com/technetwork/java/javaee/overview/index.html`
[10] `http://www.oracle.com/us/products/database/index.html`

architectural solution along with the use of ADD substantially distinguishes the systems from the majority of rivals and makes it comparable with our solution. Using a BPMS allows to easily extend and modify business processes according to business requirements. However there are also substantial distinctions compared with our solution. The systems have considerably narrower scope of applications – they are strongly oriented to business. In comparison with them (and even wider – to ERP/MRP and CRM systems in general) our framework is more universal and basic. It doesn't contain any extra and unnecessary functionality, but at the same time allows to design features quickly and with the focus on specific customer needs during the process of application development. A more basic nature of the framework form the basis for building applications for a wider class than just the commercial one, but also for management of the non-profit and public organizations, educational institutions, administrative authorities, etc. In addition, the Adempiere and Compiere systems are focused on the use of specific DBMSs. The framework presented in the paper is not tied to a particular data provider and can use any data source such as any traditional database or data center services.

4 Solution Outline

Our solution is based on a modular architecture and has exokernel as its coordinating center which controls interaction between modules. It is based on three-tier architecture that involves client processes, the central (server) process(es) and interface to the data store(s) (data stores may be external). In turn server side is divided into 3 parts: an enterprise service bus (or message bus responsible for communication), a computing part (business logic computation) and adapters for external services (including data stores, web services and so on). In addition, each module can act as a transmitter and receiver. According to the requirements of scalability and adaptability the distribution of functionality implies transferring of application functions between server and client side. Hence, both server and client sides have a set of loosely coupled modules communicating in publisher-subscriber manner. There are two types of communication in this architecture: communication within the tier (or intratier communication) and intertier one. A service can be connected to the message bus as a module (local service) or can be available via the network (remote service). Data exchange between the message bus and a remote service is managed with the adapter module. An example of the communication within the client process is depicted on figure 1. The client process has the following properties:

- the exokernel and the modules exist within the same process
- after receiving a message, the exokernel calls all registered modules
- each module is a listener of exokernel, and thus, it determines whether the module needs the message or not
- the modules keep permanent connection with the exokernel
- connection between the exokernel and modules is dual-channel: the first channel is controlling (it determines whether the data should be transferred), the second is directly for data transfer.

The intertier communication is presented on figure 2.

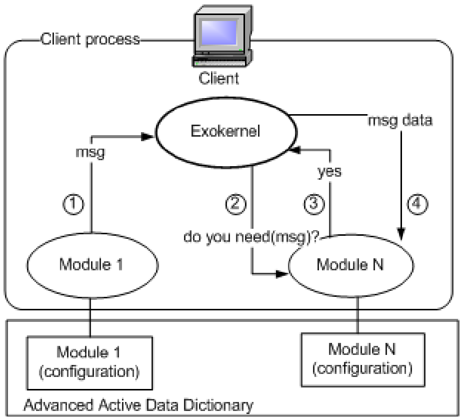


Fig. 1. An example of communication within the client process

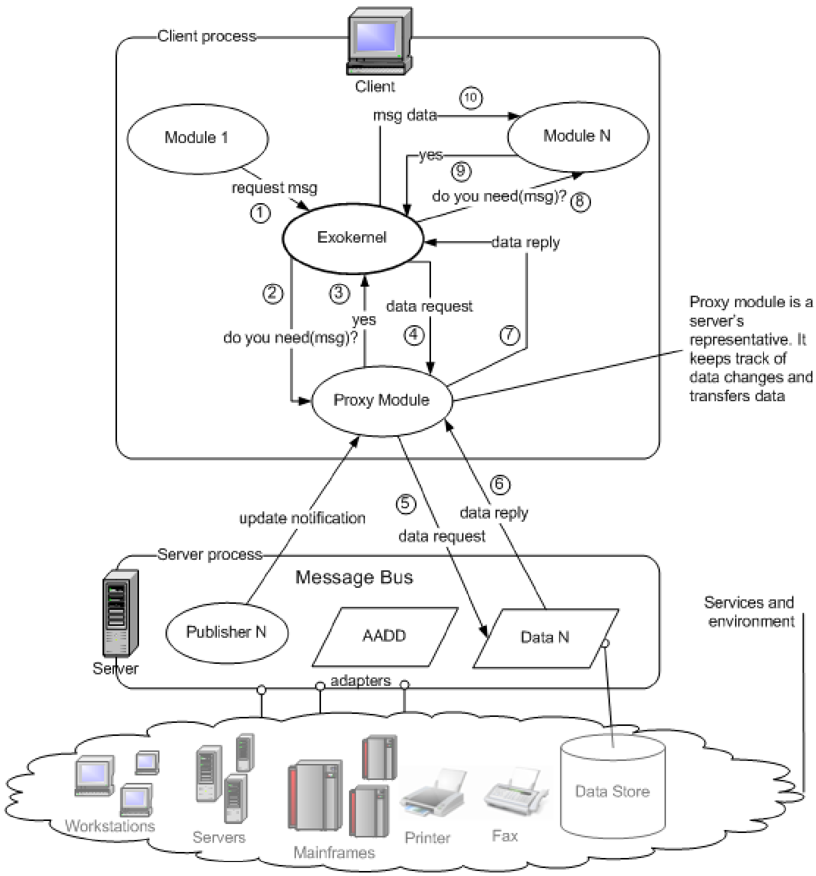


Fig. 2. An example of intertier communication

Intratier communication is intensive and fast. All modules within the tier have shared memory and therefore the main challenge here is a prompt reaction and optimal concurrent execution. The main issues of intertier communications are network traffic optimization and reliable messaging. The communication presented on figure 2 is the simplest one. It shows only a data request situation, but doesn't cover the data updating and service changing situations.

There are 3 classes of modules in the architecture:

1. GUI building modules (construct GUI from elementary blocks such as buttons, check boxes, tables, trees, etc).
2. Active record modules (data processing).
3. Active record view modules (active record and ways of displaying).

Below we summarize main advantages and drawbacks of the architecture (Table 1).

Table 1. Advantages and drawbacks of the exokernel-based modular software architecture

№	Advantages	Drawbacks
1	Functionality of a system is easily improved by increasing the number of modules	The management of a large amount of modules requires extra resources and decrease performance
2	Application life cycle is prolonging by capability of replacing particular modules for more advanced	–
3	Ability to leverage third-party plug-ins	–
4	High flexibility of application construction because of the capability of unit redeployment between the client and server tiers	Higher load on network
5	It simplifies the deployment and use of highly distributed systems	
6	All sent messages are available to all modules and each module can use or ignore it that increases the stability and flexibility of the whole system	A large number of transferring messages doesn't encourage quick performance of an application
7	Asynchrony of messages minimizes the time module spends for waiting for reply	More complicated and time-consuming coding, complexing and debugging
8	Maximally independent modules – modules can perform in different threads. Hence – the possibility of parallelization	
9	Plugging of modules on-the-fly	–

5 Conclusion

In the article we analyze the basic requirements to which an effective modern corporate system must correspond. We also provide a brief overview of the state-of-the-art theoretical frameworks that could be used to implement such class of systems and some off-the-shelf systems. We briefly describe a new solution based on the use of the concept of the Advanced Active Data Dictionary and present modular exokernel-based application architecture. In the future we plan to implement and test the proposed architecture. A detailed methodology for building of corporate systems with the use of the proposed framework is also a theme for future presentation and publications.

References

1. Mark, L., Roussopoulos, N.: Metadata Management. IEEE Computer (1986)
2. Schulte, R.W.: The Growing Role of Events in Enterprise Applications. Gartner Research, pp. 2-3. ID Number: AV-20-3900 (July 9, 2003)
3. Christudas, B.A.: Service Oriented Java Business Integration. Packt Publishers (2008)
4. Ermolayev, V.A.: Active Meta-Data Repository System for Adaptable Data Base Application Design. PhD Thesis. Zaporozhye State University (1994)
5. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley (2002)
6. Etzion, O., Niblett, P.: Event Processing in Action. Manning Publications (2010)
7. Dawson, R.E., Kaashoek, M.F., O'Toole Jr., J.: Exokernel: An Operating System Architecture for Application-Level Resource Management. In: Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles (SOSP 1995). ACM, New York (1995)
8. Narinyani, A.S.: Underdetermined systems of knowledge representation and processing. USSR Academy of Sciences (1986)
9. Narinyani, A.S. Lavrov, A.A.: The technology of active objects on the basis of virtual N-computing processors. Scientific Session of MEPhI. Collection of scientific works. At 14 v. V.3. Databases. Intelligent Systems. pp.80-81. Software, Moscow, (2001)
10. Harel, D., Pnueli, A.: On the Development of Reactive Systems. In: Apt, K.R. (ed.) Logics and Models of Concurrent Systems. NATO ASI Series, vol. F-13, pp. 477–498. Springer, New York (1985)
11. Hewitt, C., Bishop, P., Steiger, R.: A Universal Modular Actor Formalism for Artificial Intelligence. IJCAI (1973)

The Model for Enhanced Variability Management Process in Software Product Line

Olga Slabospitskaya and Andrii Kolesnyk

Institute of Software Systems of NAS, Software Engineering Department,
Akademika Glushkova st. 40, 03143 Kiev, Ukraine
{olga.slabospitskaya,andrii.kolesnyk,lnbip}@springer.com

Abstract. The paper presents a novel model for the process of managing software Variability – the ability of a software system or artefact to be extended, changed, customized or configured for use in a specific context – in Software Product Line (PL). The process pretends to be enhanced (i. e. consistent, scalable, traceable, visible and rational as for the decisions being made on Variability) to mitigate some its limitations. To this end the Model proposed composes: Management Functions uniformly combining all the actions on variability into single cycle like Doeming Plan-Do-Check-Act one; due quality Demands for the Functions; their Environment driven with another novel Model of Variability in PL. It consistently represents variability both in PL structure and artefacts across all PL development stages and stakeholders' viewpoints along with the dedicated assessment submodel purposing at the decisions' rationality. Presented sample Case Study with trial Workflow-based Configurator tool just developed in the Institute of Software Systems of NAS promises availability of the process constructed for efficient automated support.

Keywords: Software Product Line, Variability Model, Variability Management, Reusable Asset, Configuration.

1 Introduction

Effective and efficient large, complex and multi-purposed software systems composition from more simple reusable assets was one of the challenges being addressed in the research project named "Theoretical Fundament of Generative Programming and Means of its Support"(2007-2011, № 0107U002205) [1, 2] just accomplished in the Institute of Software Systems of NAS of Ukraine. Over the project Software Product Line (PL) Engineering [3] has proven to be the promising paradigm to produce a diversity of high-quality similar-but-different products with limited time and efforts.

But it was at once well-recognized that the key success factor in PL Engineering is a proper management of both the two variability types disambiguated e.g. by Metzger et al. [4]. The first is conventional software variability – i.e., the ability of a software system or artefact to be extended, changed, customized or configured for use in a specific context. In return, the second one is specific PL variability describing all the properties and qualities that should vary between the PL systems and that should not.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 162–171, 2013.

However, just now researchers' efforts concentrate foremost on variability modeling [4, 5] and implementing [3], while challengeable problems of its planning and evolving less attention are paid. One of perspective frameworks to consistently cope with them is COVAMOF [6] determining whether, when and how software variability in PL should evolve with special meta-model and method for its assessment.

The paper pursues the similar goal but for both the above types of variability and with a priori quality characteristics, namely consistency, scalability, traceability, visibility and decisions rationality, enabling the Variability Management process at hand to be enhanced. Presented Model for such a process composes its generic Functions, a priori Demands they should meet to attain the above characteristics and their common Environment with dedicated Model of Variability in PL as the base for such attaining. This model and particular Function of software variability implementation are successfully tested with trial Configurator developed over the above research project.

2 Variability Issues in PL

Variability Items to be Managed. Let fix, to use hereafter, the definitions of basic variability items that have to be manage over PL development and therefore need to be explicitly modeled, following up the origins [3, 4, 6].

The first such item is a variation point. It is an abstraction that identifies location in software system or artefact at which a choice can be made between values or variants. As Deelstra et al. [6] note, it is not by-product of design and implementation of variability, but answers the question, what does vary, being therefore identified as central element in managing variability. Each variation point is associated with a value, zero or more variants. Variation points are categorized to five basic types such as: optional (zero or one variant out of $1,\ldots,m$ associated variants), alternative (1 out of $1,\ldots,m$), optional variant ($0,\ldots,n$ out of $1,\ldots,m$), variant ($1,\ldots,n$ out of $1,\ldots,m$), and value (a value that can be chosen within a predefined range).

A variant is thus the second variability item answering the question, how does vary the variation point it is associated with [4].

The third one is dependency [3, 6]. It specifies a function of how the choices at the variation points in the PL influence a system property value, e.g. quality attribute, as well as the valid range for this value. The last one, namely constraint, is a predicate that defines possible interrelations between various variation points and variants.

Variability Levels in PL Development. Thorough study of PL Development process [2, 3] experiences five possible types (t) of variation points and variants corresponding their Life Cycle over PL Development:

– Features, i.e. abstract concepts reflecting commonalities and variabilities of software products in PL relevant for some Stakeholder that might represent a technical function, a function group or a non-functional characteristics ($t = 1$)
– Requirements as to Software Products ($t = 2$)
– Architecture components ($t = 3$)
– Database tables ($t = 4$)
– Software artefacts ($t = 5$).

Target Characteristics for Variability Management Enhancing. Based on the experience and ideas formed during the above Institute of Software Systems of NAS research project [1, 2], four Berg et al. [5] essential quality characteristics are chosen to adopt as a target for the Variability Management process to be defined. These are consistency, scalability, traceability and visibility.

Consistency means that variability should be handled the same way at all above levels of abstraction and across all PL development phases to reduce the ambiguities that might occur when using different methods for managing variability at different abstraction levels. Scalability prescribes that the methods used should be easily applicable for both the single component and a large complex system. In turn, traceability requires that Variability items at different levels of abstraction and across development phases should be explicitly linked both upwards and downwards to simplify PL evolution and maintenance. Lastly, visibility presupposes understandable representations of all Variability items in appropriate and intellectual user interface.

3 The Model Proposed to Enhance Variability Management

The model for the process at hand is proposed to form by means of its functions eliciting through comparative study of popular Variability Management templates [2-4, 6] within the perspective of Doeming's PDCA Management Cycle [7]. These functions are then improved with appropriate quality demands, their operations' unified structure and common environment, formal statements for a few mathematical problems being solved over the operations together with due solution methods.

Resulting model is a three-leveled structured tetrad of a sets open for expanding:

$$\begin{gathered} SVM = \langle \langle FN, RL, ENV \rangle; \langle AS, DM, QM \rangle; \langle CO, PM, PS, MT \rangle;\ AT \rangle; \\ ENV = \langle VM, RA, VP, RP, IP \rangle;\ \forall o_{ij} \in F_i\ o_{ij} = \langle pp, a, in, ot, [c], [pm], [ps], [m], [t] \rangle, \end{gathered} \tag{1}$$

where FN is a collection of generic Functions for Variability Management Process;

RL is a collection of FN actors' roles;

ENV is FN functions' executing environment constituted with the dedicated Variability in PL Model (VM) and relevant Repositories, namely for PL reusable Assets (RA), for Variability Profiles assessed with VM by means of special metrics from PM as described hereafter (VP), also for both the artefacts and assets Reuse Protocols () and, lastly, for interim products of PL's software systems (IP);

AS denotes a priori assumptions as to PL development organizing;

DM is a collection of Demands the Functions of FN should meet to really attain the target quality characteristics mentioned above and QM is a quality model for software systems in PL that is a priori meant as ISO/IEC 9126 model;

CO and PM denote the operations of RP's components configuring into PL's software systems and, respectively, the metrics for the process at hand;

PS and MT are respectively formal statements and solution methods for a few universal mathematical problems that should be solved over the FN's operations;

AT denotes instrumental tools the above Variability Management to support.

o_{ij} is unified structure for F_i's operation uniting: its unique purpose pp, actors' roles $a \subseteq RL$, inputs in, outputs ot and context c ($in, out, c \subseteq ENV$), specific configuring operations $c \subset CO$, process metrics $pm \subset PM$ as well as formal statements $ps \subseteq PS$ and methods $m \subseteq MT$ for mathematical problem(s) to be implemented over o_{ij} and lastly the tools ($t \subseteq AT$) to automatically support o_{ij}.

Initially AS in (1) assumes PL development to be the series of unified production rounds (where changes in the current PL systems' features and assets are prohibited and the possibility to create new features/assets is especially fixed for the round) that interchange with the rounds of PL environment reinvention (where just PL products creating is respectively prohibited).

The DM set from (1) composes the demands of consistency, scalability, traceability and variability items' visibility inspired with the title target characteristics and also additional demand of decisions rationality – i.e. obtaining each decision over FN operations together with its rational transparent for all relevant stakeholders.

Next FN set combines four target functions, namely full-informed and consistent Variability Planning (F_1) and all-aspect Controlling (F_3) with dynamic and end-round Variability Profiles and Reuse Protocols' assessment similar to COVAMOF COSVAM [6], Variability Implementing in PL artefacts (F_2) and Evolving it (F_4) up to current VP and RP. Just the last both foremost furnish due rationales for appropriate managerial decisions over FN processing.

Additionally, FN includes Variability Management Initiation (F_0) – the fifth service function to create all the necessary technological prerequisites as well as initial VM and RA consistent with it.

In turn, main roles in RL set should be Manager (for PL, product portfolio, release, asset etc.), Designer (business analyst, architect, promoter etc.), Developer (programmer, integrator, system administrator etc.) and, last but not least, Quality Assurer.

Based on some Variability Management templates [2-4, 6] comparative study three subsets should be proposed for PS and MT sets, last in SVM (1), such as Optimization on Graphs statements and methods (for F_0, F_1), expert-analytical assessment [9] (for F_1, F_3) and also Quantitative and Qualitative Data Mining ones (for F_4).

It is worthwhile to note that the constructive Core of SVM (1) is thus $\langle FN; VM \rangle$ couple that drives, more or less directly, the other SVM elements' structure.

4 Consistent, Traceable and Visible Model for Variability in PL

Let's particularize the title quality characteristics and demands DM from SVM (1) to fix inspired demands the dedicated VM should meet to pursue its purpose in (1):

– uniform, consistent and traceable representation for all the variety of variability items and their interrelations over all the stages both for PL Domain and Application

Engineering processes [2-6, 8] as well as for all the PL stakeholders' viewpoints over all functional segments from PL scope

— using traceable notations for PL artefacts modelling appropriate to their types

— identifying explicitly commonalities and variabilities across all PL development artefacts, stages and stakeholders' viewpoints

— full-informed, unified and consistent assessing of Variability Profile.

Relevant Model for Variability in PL is defined [8] to be a hybrid structured triple

$$VM = \langle SV; AV; EV \rangle, \tag{2}$$

where submodels SV and AV represents variability in the PL structure and in its artefacts corresponding to above-mentioned PL variability and software variability;

EV is an integrated submodel for informed and consistent variability assessment.

The first submodel SV in (2) gives the formal representations of all the features from PL scope, both commonalities and variabilities, artefacts to implement them and their links on the base of feature modeling approaches [2-4, 6]. It is a structured tuple

$$SV = \langle G_1; \langle \langle G_t, TR_t \rangle, t = 2,3,4,5 \rangle; CN; DP \rangle, \tag{3}$$

where $G_t = \langle F_t, LF_t \rangle$ is the graph where the nodes (F_t set) are unique identifiers of the above t-typed PL artefacts (features, requirements, architecture modules, data elements, tests, reusable components and software systems) that are linked through the relations of obligatory and variant binding (LF_t set);

TR_t is bilateral traceability links between the nodes of G_{t-1} and G_t graphs;

CN , DP are the predicates on $\otimes_{t=1,\dots,5} F_t$ for PL constraints and dependencies.

In turn, the second submodel AV in (2) provides unified formal representations for all PL software products currently located in repository RA, being developed or might be developed eventually within current PL scope together with their development products (requirements etc.). To explicitly put the elements of SV (3) into particular PL's software system, any t-typed artefact is formally seen as cross-cutting "fragment" of SV . It is bounded with continuous upwards – downwards traceability links TR_t from (3), which interrelates this artefact with the features it should implement and the final software product. The model AV is a structured tuple

$$AV(id_t) = \langle g_1; \langle \langle g_u, tr_u \rangle, u = 2,\dots,t \rangle; \langle \langle p_u, tr_u \rangle, u = t+1,\dots,5 \rangle; cn; dp; s \rangle, \tag{4}$$

where id_t is the modeled artefact's unique identifier;

g_u and p_u are subgrafs of G_u from (3) representing the artefacts that are implemented with id_t and, respectively, implement it over PL development;

$tr_u \in TR_u$ are subsets of traceability links between the nodes of g_{u-1} and g_u ;

cn and dp are the limitations of CN and DP on Cartesian product of g_u and p_u nodes sets and s is the artefact's current state (e.g. "core asset" etc.).

Note that each of the five "horizontal" planes, implicitly defined with formulas (3), (4), reflects the viewpoints both at PL and artefact variability by particular PL Stakeholders group being represented over PL development with the proper-typed artefacts – from the customers' features at the first level (G_1, g_1) downwards to the programmers' and testers software and tests at the fifth one (G_5, g_5).

An assessment submodel EV, the last in VM (2), is proposed to form in accordance with its predecessors SV (3) and AV (4) through all-leveled commonalities eliciting from SV similar to Metzger's Orthogonal Variability Model forming [4]. The residual three-dimensional graph is further divided into separate "orthogonal" subgraphs. These subgraphs are constituted with all the PL variation points (P) and, respectively, variants for them (V) together with the saved binding and traceability links from SV.

But now both the new subgraphs P, V are furthermore split into five blocs corresponding to the above types of PL artefacts – SV's levels $t = 1,...,5$ from (3). Each t-leveled block unites appropriate subgraphs of variation points $P_t \subset P$ and their variants $V_t \subset V$. Dimensions of P_t and V_t are additionally provided with the third dimension of full-informed quantitative variability estimates at the bloc's level t in some relational scale (e.g. Saathy's fundamental one). Two types of such estimates are proposed, namely Autonomous Variability Index (ai_t) and Variability Compliance one with both the PL customers' business needs in its products and developers' requirements for them (ci_t). Both indexes pentads are proposed to obtain with an expert-analytical methodology [9] based on the bloc's data about variation points, variants and their links (both binding and traceability) as well as actual data from RA, IP and RP repositories from (1) and to integrate in corresponding Indexes.

The structure proposed of resulting integrated model is depicted with Fig. 1.

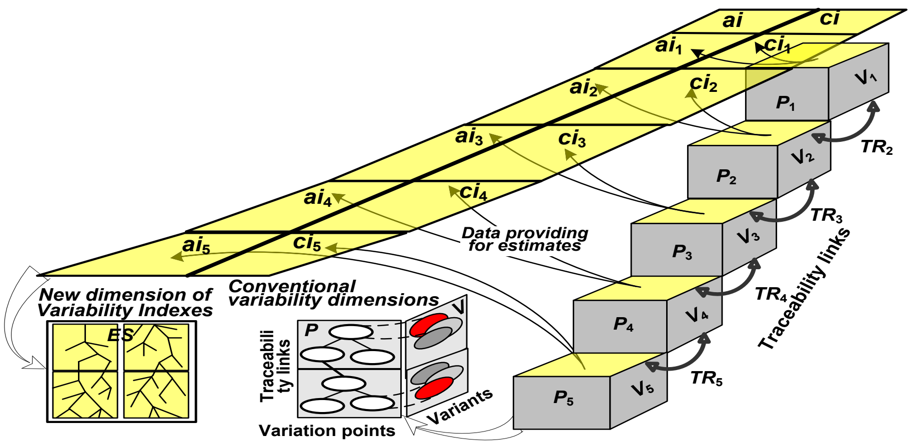


Fig. 1. The structure of Integrated three-dimensional variability assessment submodel

This model is formally the five-leveled triple

$$EV = \langle\langle P_1, (P_t, TR_t), t = 2,...,5\rangle; \langle V_1, (V_t, TR_t), t = 2,...,5\rangle; \langle ai; ci; (ai_t, ci_t), t = 1,...,5\rangle\rangle, \qquad (5)$$

where ai and ci denote above-mentioned Integrated Indexes of Autonomous Variability and Variability Compliance as the roots of hierarchical submodels proposed ; ai_t, ci_t are these submodels' t-leveled fragments reflecting t-typed PL artifacts.

To represent the fragments $\langle ai, (ai_t, t = 1,...,5)\rangle$ and $\langle ci, (ci_t, t = 1,...,5)\rangle$ from (5) purposing at the estimates' plausibility and consistency to increase, universal preferences model such as Bayesian Net, Saathy's Analytical Hierarchy and von Winterfeldt's Value Tree with the common set of terminal nodes are proposed to be configured up to the current assessment situation [8] over PL development. In general, for individual forecasting under high uncertainty (at the beginning of the above production rounds) Bayesian Net is the most appropriate while for collective assessment (at their ends) Value Tree is.

Current structured estimates of ai, ci with (5) filled up with t-layered estimates of the efforts for t-typed artefacts creation and assets reuse intensity, should be stored in Variability Profiles $vp \in VP$ (see formula (1)).

5 Case Study to Test the Model's Core

Here the probe implementation of PL artefact's variability model AV (4) and $F_2 \in FN$ (1) function of Variability Implementing in PL artefacts are considered for sample "toy" domain of quadratic equations solving. While classical feature diagram [4, 6] fairly clarifies variability items at the feature level, it's quite difficult to use it at the lower AV levels. Instead MS Visual Studio Windows Workflow Foundation (WWF) may enable more details about AV with appropriate diagram (see Fig. 2).

Let's explain how such variability might be visualized and on-line managed with WWF diagram. Note that "*A*" letter at the Fig. 2 connected with the "plus" pictograms denotes the variation points that might be bound with reusable components as their variants. In turn, "*B*" letter denotes such variants. In the case at hand the "Discriminant" component contains simple code to find discriminant ($D = b * b - 4 * a * c$), implemented by the developer responsible for producing PL core assets. Depending on its operating outcome, there are three scenarios (use case): $D > 0; D = 0; D < 0$ and corresponding components for them: "TwoRoots", "ExactlyOneRoot", "NoRoots".

With WWF Visual Studio environment don't prohibit the developer from binding variation points with any reusable components. That's why a dedicated software tool is needed that should support both the VM proposed (2)-(5) forming and actualizing and application configurations changing based on VM and repository's assets.

Trial prototype of such a tool, named Configurator, is implemented within the Tool Suite for Software Systems Development just elaborated in the Institute of Software Systems of NAS [2, 10]. It purposes at configuring and modifying a diversity of similar-but-different applications from reusable components through filling them up with variation points and variants [1, 2, 8] based on WWF [2, 10, 11]. An interim result of components' building with Configurator is XML file shown beneath. It is an instruction for executable file compiling to create the target application.

Configuration process producing this XML file is initiated through the special chart processing with embedded WWF tool in Visual Studio environment (see Fig. 3).

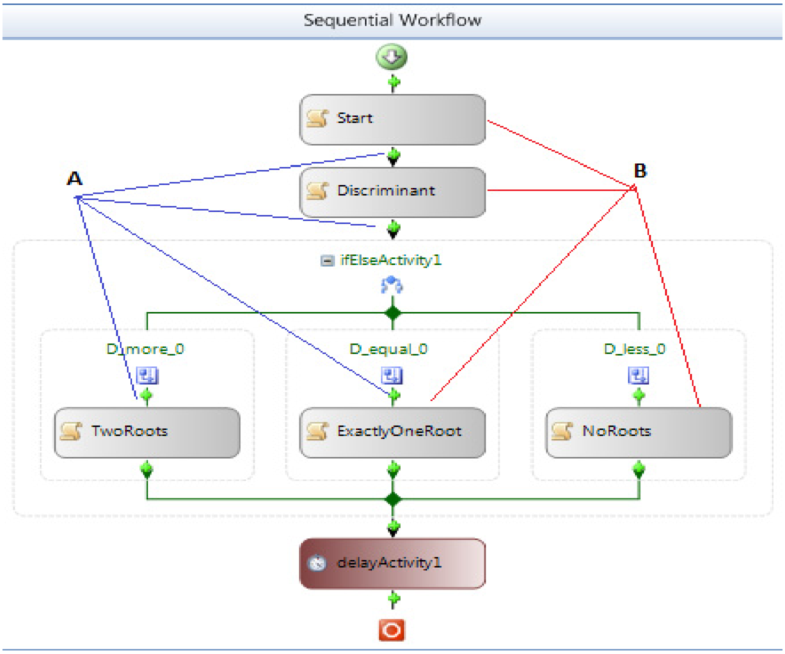


Fig. 2. Sample artefact variability model and reusable components are represented

```
<SequentialWorkflowActivity>
         <CodeActivity x:Name="Start"
ExecuteCode="codeActivity1_ExecuteCode" />
         <CodeActivity x:Name="Discriminant"
ExecuteCode="codeActivity1_ExecuteCode_1" />
         <IfElseActivity x:Name="ifElseActivity1">
         <IfElseBranchActivity x:Name="D_more_0">
         <IfElseBranchActivity.Condition>
         <CodeCondition Condition="WorkMeth1" />
         </IfElseBranchActivity.Condition>
         <CodeActivity x:Name="TwoRoots"
ExecuteCode="codeActivity1_ExecuteCode_2" />
         </IfElseBranchActivity>
         <IfElseBranchActivity x:Name="D_equal_0">
         <IfElseBranchActivity.Condition>
         <CodeCondition Condition="WorkMeth2" />
         </IfElseBranchActivity.Condition>
         <CodeActivity x:Name="ExactlyOneRoot"
ExecuteCode="codeActivity2_ExecuteCode" />
```

```
          </IfElseBranchActivity>
          <IfElseBranchActivity x:Name="D_less_0">
          <IfElseBranchActivity.Condition>
          <CodeCondition Condition="WorkMeth3" />
          </IfElseBranchActivity.Condition>
          <CodeActivity x:Name="NoRoots"
ExecuteCode="codeActivity3_ExecuteCode" />
          </IfElseBranchActivity></IfElseActivity>
          <DelayActivity TimeoutDuration="00:00:05"
x:Name="delayActivity1" />
</SequentialWorkflowActivity>
```

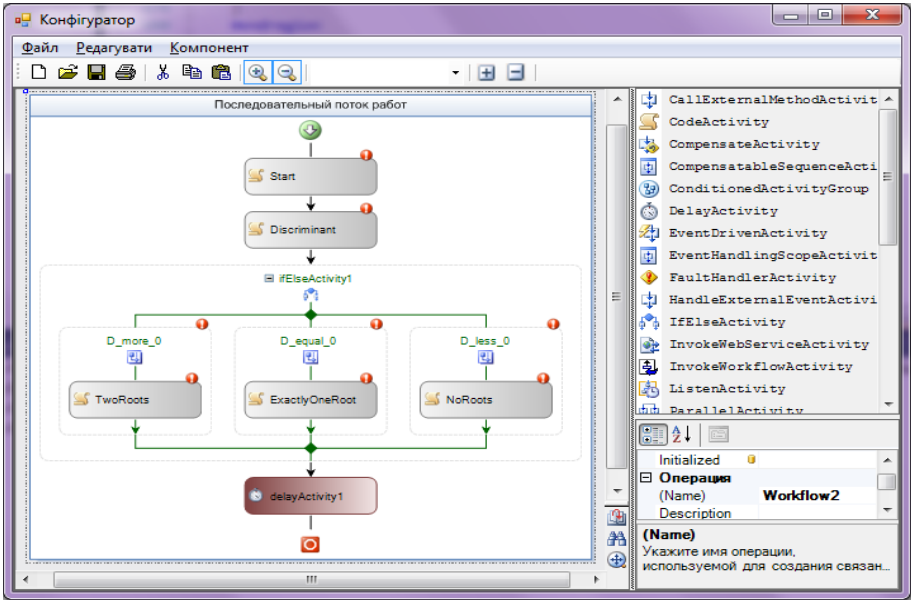


Fig. 3. Processing the reusable components by the Configurator is depicted

Note that the application created is variable i.e. enables square equation solving under various conditions prescribed.

6 Conclusions

A novel Model is elaborated for enhanced (i.e. consistent, scalable, traceable, capable to visualize variability items in PL and provide rational decisions on it) Variability Management process. The Model's core unites generic Management Functions (with actors' Roles and unified operations) and another novel Model for Variability in PL that drives the Functions' executing Environment. The Core is filled up with target quality Demands for the Functions, a priori Assumptions about PL development, formal Statements and solution Methods for a few universal mathematical problems that should be solved over the operations.

To meet the Demands Variability Model is proposed to uniformly and consistently represent all its items over all abstraction levels both in PL and in the PL artefacts and also across all the relevant stakeholders' viewpoints. Moreover, it includes dedicated submodel for rational, full-informed and consistent variability assessment.

In turn, four target Functions are prescribed such as full-informed and consistent Variability Planning, Controlling it through continuous rational assessing based on the above submodel, Evolving up to retrospective and current assessment results and, lastly, Implementing in PL artefacts. These functions are serviced with the Initiation one to create Variability Model's frame and technological prerequisites for them.

Promising Case Study is drafted of product variant deriving in sample domain with trial Workflow-based Configurator developed in the Institute of Software Systems of NAS of Ukraine just to test the above-mentioned Model's core.

Based on it, the authors believe the Model presented is constructive for Enhanced Variability Management process to integrate in the PL development that would efficiently balance all the PL stakeholders' demands for variability as well as for agile and flexible structure of supporting services to elaborate for such a process.

References

1. Lavrischeva, E.: Generative Programming of Software Products and Their Families (in Ukrainian). In: Problems in Programming, vol. 1, pp. 3–16. Kiev (2009)
2. Lavrischeva, E., Koval, G., Babenko, L., Slabospitska, O., Ignatenko, P.: New Theoretical Foundations of Production Methods of Software Systems in Generative Programming Context (in Ukrainian). In: Electronic Monograph, UK-2011, vol. 67. Kiev (2011)
3. van der Linden, F., Schmid, K., Rommes, E.: Software product lines in action: the best industrial practice in product line engineering. Springer, Heidelberg (2007)
4. Metzger, A., Heymans, P., Pohl, K., Schobbens, P.-Y., Saval, G.: Disambiguating the Documentation of Variability in Software Product Lines: A Separation of Concerns, Formalization and Automated Analysis. In: 15th IEEE International Requirements Engineering Conference, pp. 243–253. IEEE Press, New York (2007)
5. Berg, K., Bishop, J., Muthig, D.: Tracing Software Product Line Variability – From Problem to Solution Space. In: Proc. of SAICSIT 2005, pp. 111–120 (2005)
6. Deelstra, S., Sinnema, M., Bosch, J.: Variability assessment in software product families. In: Information and Software Technology, vol. 51, pp. 195–218. Elsevier (2009)
7. Walton, M.: The Deming Management method. Dodd, Mead. New York (1986)
8. Lavrischeva, K., Slabospickaya, O., Kolesnik, A., Koval, G.: The Theoretical View for Software Family Variability Management (in Ukrainian). In: Bulletin of University of Kiev. Physics & Mathematics, vol. 1, pp. 151–158. Kiev (2011)
9. Lavrischeva, E., Slabospitcka, O.: An Approach for Expert Assessment in Software Engineering. Cybernetics and Systems Analysis 45(4), 638–654 (2009)
10. Lavrischeva, E.: Instrumental and Technological Complex for Developing and Learning Aspects of Software System Development (in Ukrainian). In: Bulletin of NAS of Ukraine, vol. 3, pp. 17–27. Kiev (2012)
11. Kolesnik, A.: Approaches to Configure Reusable Assets (in Ukrainian). In: Problems in Programming, vol. 4, pp. 63–71. Kiev (2011)

Providing the Interactive Studying of the Students Which Is Controlled by Distance with Web-Portal E-olimp

Borys Lyashenko, Sergiy Zhykovskyy, and Svitlana Postova

Zhytomyr State University named after Ivan Franko, Computer Science Editorial,
Velyka Berdychivska, 40,1008 Zhytomyr, Ukraine

Abstract. In this paper we review studying informational technologies of the interactive education which is controlled by distance. We were proposed to create and apply WEB-portal (www.e-olimp.com.ua) for realization of distant interactive educational studying. We explain that using of given Web-portal will permit to extend the received informational technologies on other disciplines and specialties will help to save finances on establishment and refreshing the educational, educational and methodological literature routing its location in corresponding web-books on it.

Keywords: distant interactive studying, interactive technologies, Web-portal e-olimp.

In our days access to Internet is possible mostly to every student, and the amount of Global Net (WEB) users is growing every day. So fast evolution of the Internet brought to his intromission to the most of the sphere of human activity. It applies to the education and science too. Among the traditional science paper issue exist also web (non-legible) issues, less part of what is situated in web-space.

Occurs the enlargement of science sphere of WWW (as in branch-thematically, so as in national-geographical plan) and, as circumstance, growth of popularization and meanings of various form of distant studying. Occurs placement of web samples of studying literature in WEB (NET), at the same time as plenty of modern books, magazines, issues created in both versions – paper and numeric (digital).

However great amount of researchments concerning questions from organization of professional students youth qualification is badly opened, specially questions with using modern informational technologies. It is an actuality of our reseachment. In summary the aim of our article is studying informational technologies of the interactive education which is controlled by distance.

Questions connected with usage of web in studying and science activity, were taught by the method of general usage of directions in Internet technologies in higher educational establishments (Yu. Ramskiy, B. Mokin, V. Grabko, V. Mesyura, S. Yuhimchuk, O. Shlikova) and Internet-literacy of students (A. Taha).

The process of the informatization became regular factor of the IT-penetration of the studying. Unfortunately, in our days the degree of the informatization of the

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 172–177, 2013.

education and society in the Ukraine is rather less than in other developed countries. One of the main tasks of the IT-penetration of education is the implementation of the educational instruments and effective using of the existed and always developed resources of the informational-communicative technologies, organization of the dynamic net interaction of every members of the studying process.

Therefore in the base of the Web-portal of the organizational and methodical supply of the Olympiad from programming which is controlled by distance E-olimp were created the department of the interactive studying of students which is controlled by distance for professional training of the students from specialties "Informational and communicative technologies" and "Computer Sciences" [1].

Actual task of modern (contemporary) studying system is the realization of qualified and personal oriented usage in studying. Question of the activization of studying process is concerning to the most important problems of modern pedagogical science and practice. Realization the foundational activities in studying has its own meanings, studying and development has revealing character depends from quality of studying, derivation and professional skills of the students.

Contemporary books, periodic for tutors permit under necessary practice build process of studying such way that gives the possibility to develop students intellection, attraction and other features of cognitive activity. Productive studying must form not only deep notion, and skills to use them in various situations, individually to get knowledge, form to solve problems. Best results can be aimed only with the help of active position of students in educational process.

The main methodical innovations are associated with applying of the active methods of studying (interactive). Interactive studying first of all is a dialogical learning in the way of that is realized the team-work.

Problems of definition, explanations of implication in interactive technologies and usage of them in studying and educational process were made by V. Kotov, L. Pyrozhenko, O. Pometun, O. Savchenko, O. Yaroshenko etc. Particulary O. Pometun developed and promulgated the technology of the interactive studying in Ukraine [2].

As known, interactive methods are methods which let us to share knowledge with each other. Interactive studying is studying built on interaction of all members of studying, including tutor (scholar). These methods most of all correspond to personal oriented concept it allows co-education (collectively, studying in cooperation). In such case tutor is a organizator of learning process and leader of the group etc.

The researchments made in 80-s by National Training Center (USA Maryland) performed that interactive methods let us quickly increase the per cent of understanding the information.

Place of passive, active and interactive methods also is very good displayed in - "Bloom's taxonomy" – it is a table of levels and aims of education and development of cognitive skills, made by group of American psychologists and educators headed by prof. Benjamin Bloom [3]. B. Bloom suggested hierarchy of educational aims over their difficulty. According to Bloom's taxonomy passive methods of studying is oriented, as rule, only on levels of knowledge and understanding, interactive - works with all levels of education. But merely, this is not obliged to use only interactive methods.

Consequently, interactive studying is a special form of the organization of the cognitive activity which main goal is creating of the comfortable specifications of studying when every student feels his own success rate and intellectual ability [4].

The interactive studying at the same time solve some tasks:

- develops communicative skills and abilities, helps to fix emotional contacts;
- solves informative task because it provides students with necessary information without it is impossible to realize common activity;
- develops general skills and abilities (analysis, synthesis, formulation of aims etc.), rather it provides solving the tasks;
- provides the educational task, because it habituate for teamwork, listen the opinion of other people.

The interactive studying partially solves one essential task: relaxation, getting rid from mental load, switching attention, changing activity forms etc. One variety of modern interactive studying is interactive distant studying. Modern educational sphere understands under distant studying – complex of educational facilities (utilities) which is given to great variety of population (people) in our country and abroad with the help of specialized informative-educational environment based on exchanging with information on distance: satellite TV, radio, net connection etc.

Distance studying – cooperation of teacher and student with each other on distance, and reflects all possible educational components (aims, plot, methods, organizational forms, methods of studying) and often realized with the help of special methods of Net technologies or other ways, which foresight interactivity.

Modern distant education is based in using next primary elements: methods of delivering the information, methods, depends from technical environment of exchanging information. Now the main perspective is interactive cooperation with students with the help of informational communication networks out of them relieves the location of network users.

Distant studying let us: reduce costs for realization of education (there is no need to spare costs for rent, voyages to place of studying as students as teachers also); perform studying for great amount of students; upgrade quality of education using modern resources, web libraries; create the unique environment. Additionally distant education gives students an opportunity to study in any time and any temp they choose. In distant education all studying process can be uninterrupted.

As a result, we were proposed to create and apply Web-portal for realization of distant interactive educational studying on the example of normative disciplines as "Programming", "Theory of Algorithms", "Theory of Programming", "Artificial Intellect", "Discrete Mathematics" and specialties "Informational-Communicational technologies " "Informatics" On created web-site we plan to place online tutorials and manuals on selected subjects, instructive-methodological materials for realization of laboratory studying, and also individual tasks with various difficulties concerning students skill level, summary module control works and works according sections, made for self processing.

Based on given portal also it is foresighted independent (computer-generated) check of theoretical skills and practical abilities from chosen subjects, leading the periodical with automatically determination of ratings of every student, discussing problems online revealing during study process.

The created Web-portal can provide:

- organization and realization of the intramural studying and studying which is controlled by distance from chosen normative subjects;
- independent (computer-generated) check and analysis of students results from normative subjects such as Programming, The Theory of the Algorithms, The Theory of Programming, Artificial Intelligence, Discrete Mathematics and for specialties "Informational and communicative technologies" and "Computer Sciences*";
- organization and realization of student Olympiad (contests) from elementary level to all Ukrainian as in intramural and distant variants;
- upgrading the professional students qualifications;
- popularization professional skills of students youth to programming professions.

In foundation of these project is settled Web-portal organizationaln and methodological supplyment distance contests (Olympiads) from programming for talented young people from Ukrainian Institutions which was at Zhytomyr State University named Ivan Franko with financial support in Educational and Scientific Ministry of Ukraine in 2009-2010 years (www.e-olimp.com.ua). This site was created for support and development from the sport programming.

The site E-olimp is supported in 3 languages (Ukrainian, Russian and English) what gives us possibility to engage members from different countries. During the time system is existing in Internet (five years) it is registered more than 12000 of users from Ukraine, more than 1000 users from abroad (Russia, Kazakhstan, China, Korea, Iran, Turkey etc.). During this period were sent more than 600000 of solves, which were immediately checked by system.

During the creating E-olimp system for the authors of the project was task to make it accessible to a large audience, easy to use, quick operation, that is why the system check was presented as a site found on the Internet, and include opportunities:

- locating tasks with the help of simple tool bar;
- compilation of solves and its testing;
- performing competitions, in located terms;
- rating statistics for monitoring tests results;
- scanning the testing results as whole and separately according every test;
- start the testing program as an application in OS based on Windows NT platform;
- discussion of topics, concerning competitions, programming, forum etc.

The Web-portal consists from 2 modules:

first – **web**-site, developed in PHP framework, where users can read tasks conditions, send their solves and get through results, discuss conditions, system work, give their comments according site topics;

second site was created for security and speed work, and written on C++, could be installed on other server.

Both modules are united with server data base (DB) Program is using DB-MySql server (Fig. 1).

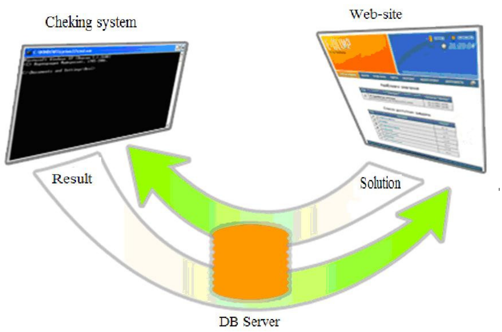


Fig. 1.

Web-site is appropriated for displaying announces about competitions, proposed texts of tasks, testing results got from solved and published competition results. Checking system provides the compilation of solves, check and posting the results to data base (Web-library). Every willing have a possibility to take part in competition, which is held on E-olimp, or merely check its own solves of tasks, conditions of what is located in data base of the site [5].

Such way, realization of such project supports in: the improvement of quality concerning University and school nature-mathematical education, strengthening the educational-methodical and economical-technical base in the educational institutes; modernization the systems of training and upgrade qualifications of the pedagogical stuff: development of science researchments in branches of intellectual informational technologies of studying, which farther will assist in development of innovation economics based on the integration to educational, science and productive activity.

Among that, using of given Web-portal will permit to extend the received informational technologies on other disciplines and specialties will help to save finances on establishment and refreshing the educational and educational-methodological literature routing its location in corresponding web-books on it. Installation various kinds of studying from Informatics on given Web-portal will allow user to get free of charge Web-books and instructional-methodological materials for using them in laboratory and practical tasks. Students (and pupils) will have an opportunity to perform laboratory and practical studying, personal and individual tasks with the help of Web-portal and to direct them to net check with further ability of getting independent tutors object mark.

References

1. Lyashenko, B.M.: The Internet-portal of the Provisionof olympiad from programing which is controled by distance. In: Lyzshenko, B.M., Zhukovskyy, S.S. (eds.) The Informational Technologes on education: Digest of Scientific Works, vol. (4), pp. 134–138. KhSU, Kherson (2009)
2. Byelyenka, G.V.: The Interactive methods in Modern Pedagogy. In: Byelyenka, G.V., Rasova, O.S. (eds.) Humanitarian Heraldvisnyk SHEI «Pereyaslav-Khmeknyckyy State Pedagogical University named after Grygoriy Skovoroda»: Digest of Scientific Works, vol. (21), pp. 33–38. Pereyaslav-Khmelnyckyy (2011)

3. `http://en.wikipedia.org/wiki/Bloom’s_Taxonomy`
4. Lopuhina, V.: The Tehnologies of the Children’s Studying, Training and Development. In: Lopuhina, T.V., Styagunova, O.O. (eds.) Studying and Methodical Textbook, 146 p. Kashtan, Doneck (2010)
5. Zhukovskyy, S.S.: The Organization of the Preparation of the Pupils olympiady from computer Scinces by Means of the Internet-portal E-olimp. In: Zhukovskyy, S.S. (ed.) The Informational Technologes on education: Digest of Scientific Works, vol. (4), pp. 202–206. KhSU, Kherson (2010)

A Template-Based Method to Create Efficient and Customizable Object-Relational Transformation Components

Igor Lihatsky, Anatoliy Doroshenko, and Kostiantyn Zhereb

Institute of Software Systems of National Academy of Sciences of Ukraine,
Glushkov prosp. 40, 03187 Kyiv, Ukraine
igor_md@ukr.net, {doroshenkoanatoliy2,zhereb}@gmail.com

Abstract. We describe a method to create object-relational transformation components by using the code generation system to automatically generate a persistence layer based on the database structure. A text template engine is used to generate SQL queries, business classes and APIs to access data from the application code. Provided default implementations are sufficient to quickly obtain a working persistence layer. However the real power of proposed solution lies in extensive customization capabilities. Therefore the developed system provides a high developer productivity because of automation, as well as high performance and flexibility due to the possibility of customization. Performance measurements demonstrate the high efficiency of the generated code, both in terms of execution speed and code size.

Keywords: object-relational paradigm mismatch, persistence layer, code generation, text template engine, object-relational mapping.

1 Introduction

Currently most business applications are created using object-oriented languages such as Java, C#, C++, and store persistent data in relational databases such as Oracle, Microsoft SQL Server or MySQL. Both object-oriented and relational paradigms provide a well-established, reliable foundation for solving applied problems. They are supported by numerous tools and methodologies that simplify the development process and improve the product quality.

However, the underlying data models for object-oriented and relational systems differ significantly. Therefore applied developers face the need to implement a conversion between object-oriented and relational representation of the same data. Sometimes such conversion is implemented manually, in which case obtained code is more efficient, but developers need to perform quite significant routine work of manually transforming each database object. In other cases automated tools (such as ORM systems [9]) are used, improving the developer productivity, but often sacrificing runtime performance and flexibility. There is a need of solution that can combine the flexibility and performance of manual implementations with automation achieved by using ORM systems.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 178–184, 2013.

In this paper we describe our approach to object-relational transformation by using the code generation system to automatically generate a persistence layer based on the database structure. A text template engine is used to generate SQL queries, business classes and APIs to access data from the application code. Provided default implementations are sufficient to quickly obtain a working persistence layer. However the real power of proposed solution lies in extensive customization capabilities. Therefore the developed system provides a high developer productivity because of automation, as well as high performance and flexibility due to the possibility of customization. Performance measurements demonstrate the high efficiency of the generated code, both in terms of execution speed and code size.

The contribution of this paper is not some absolutely new technique or method, but rather a new combination of existing methods, such as using code generation and text templates, in order to improve the runtime efficiency of object-relational transformation, as well as increase the developer's productivity.

The rest of the paper is organized as follows: first we briefly analyze the current state of object-relational mismatch problem and describe existing solutions. Then we discuss our approach in detail, describe our implementation of this approach, illustrate capabilities of the developed system on simple examples and evaluate the performance of our system. Finally we provide the conclusions and directions of future research.

2 Object-Relational Mismatch: Problems and Approaches

Currently most business applications are created using two mainstream programming paradigms: object-oriented programming (OOP) [3] and relational database management systems (RDBMS) [2]. However these two paradigms use different and incompatible data representations: an object graph in OOP and tables in RDBMS, and the data access approaches also differ significantly. Because of such differences between OOP and RDBMS (often referred to as a *paradigm mismatch*, or an *impedance mismatch* [9]) there is a need of components that perform transformation between these representations. While developing such components, two main goals are reducing runtime overhead for each transformation and reducing development effort.

One approach to connect OOP and RDBMS representations is hand-coding the persistence layer. In this case the developer has to manually implement saving and loading data using SQL queries and low-level database access APIs (e.g. ADO .NET for Microsoft .NET framework or JDBC for Java). Advantages of this approach include maximum performance, complete control over the process of saving and loading data and a possibility to use advanced features of particular database systems. The main drawback is a large amount of routine and error-prone work, during both implementation and maintenance of the persistence layer.

Another approach is to use object-relational mapping (ORM) systems that automatically save and restore the object graph into RDBMS using a formal description of a mapping [1]. Popular ORM solutions include Hibernate for Java [1], NHibernate and Entity Framework for .NET [8]. The advantages of ORM solutions include significantly reduced development efforts, increased maintainability and independence from the database provider. Their main drawback is the runtime overhead: automatically generated SQL queries can be less efficient compared to those created and optimized by hand. Also ORM systems can generate unused features, such as queries for updating a table that should be read-only. Some ORM solutions may impose restrictions on the domain classes, such as inheriting from the base class provided by ORM or restrictions on the types of collections and associations.

Yet another approach to avoid a paradigm mismatch is to use object-oriented or object-relational databases [11]. Such solutions include a direct support for object-oriented features such as the object composition, link navigation, encapsulation, inheritance and polymorphism. Popular object databases include Caché [6] and Google App Engine Datastore [10]. Also object-oriented features are included in traditional relational databases, such as Oracle and Microsoft SQL Server [11]. By using object databases the transformation overhead can be avoided entirely, thus increasing both the developer productivity and the runtime performance. Another advantage is a simpler data model that is common for the data storage and data manipulation. Major drawback of this approach is reliance on less popular, and therefore less optimized and standardized database engines. Therefore currently object databases are used in some specific applications [11], but they cannot replace RDBMS as a mainstream data storage technology.

The analysis shows that current solutions of the object-relational paradigm mismatch force the application developer to make a choice between the automation and flexibility. Therefore our goal is to find effective ways to store the object graph aimed at achieving maximum performance and flexibility, as well as the sufficient degree of automation.

In this paper we use the following approach. We create the custom persistence layer, similar to the manual approach, by using code generation tools to automate its creation and reduce amount of routine coding. RDBMS are used as the data storage (instead of object databases) because of their high reliability and performance. The generated persistence layer consists of business objects (classes) that interact with the database and are generated from the database structure. The persistence layer provides safe interaction with the database by default, to prevent attacks such as SQL injections. Persistence layer operates with strongly typed objects that allow catching many errors and typos during compilation and not during query execution. All aspects of generated code can be customized without changing the structure of the application.

There are existing systems that use the similar approach, such as a NetTiers Application Framework [7]. However it is based on a commercial CodeSmith template engine [5] which increases the cost of the system. Also the NetTiers system does not generate strongly typed objects for all stored procedures, and supports only C# language – there is no possibility to generate the code in different programming languages.

3 The C-Gen System

In this section we describe the C-Gen – our code generation system that can be used as a solution of the object-relational paradigm mismatch.

The C-Gen system generates a persistence layer of the application using a database structure as an input. The database describes the entities in a subject domain and therefore can be used to create business objects. Currently the system is unidirectional: we can generate business objects from the database structure, but not vice versa.

The C-Gen uses a code generation approach, i.e. the automatic generation of source code from the given input data. As a code generation tool we use a Text Template Transformation Toolkit (T4 [4]). Templates are used to generate a program source code based on the model (database structure). The generated file can use an arbitrary text format, in particular, it can be a program source code in any language.

The generation process starts from existing database that is designed manually. A domain model is represented as a set of tables and views with relations. The database developer takes all responsibility for creating database. This approach supports maximum performance and flexibility of the created application. The developer has a full control over the process of designing and creating the database. As a result the developer can create a high-quality and efficient SQL code.

When the domain model is implemented in the database, we use the C-Gen system to generate a persistence layer. It consists of stored procedures inside the database, as well as business objects and access methods in a source code. The C-Gen supports two types of stored procedures: *simple stored procedures* support CRUD (create, retrieve, update, delete) operations and are generated automatically; *custom stored procedures* are specific to a concrete situation, and their design is fully controlled by the developer. Thus, by generating simple stored procedures automatically the developer is relieved from writing them by hand. On the other hand we reserve the possibility of implementing performance-critical stored procedures for the developer.

Custom stored procedures are also used to represent the relations between the tables. The tables can be related in one of three different ways: one-to-to one, one-to-many, many-to-many. Therefore one object in OOP can be represented as several connected RDBMS tables. There are other differences between RDBMS and OOP data model: tables in RDBMS don’t support encapsulation, inheritance and polymorphism which are basic properties of OOP paradigm; tables have explicit identity provided by primary keys while objects can rely on implicit identity of memory location; objects support link navigation while tables rely on joins to access related properties.

Custom stored procedures that represent the relationships between the tables and views contain logic of joining the tables and views between each other. As the result we have a new object that represent such relation. This object presented in souse code as strongly new class, and custom procedure as the method. In this way we can map database relations on objects.

The next step after generating stored procedures is creating business objects. For each entity in the database, the C-Gen system generates a corresponding class. Each database field is represented as a strongly-typed property. For each custom stored procedure the C-Gen system also generates a corresponding class based on the procedure name and return fields. The final step is the generation of access methods for simple and custom stored procedures; their parameters and return type depend on stored procedure signature.

After the C-Gen system completed its work, the application developer can work with business classes as required in application. Working with database is completely hidden behind generated methods. If some changes are made to the database structure, the persistence layer can be regenerated. Therefore the C-Gen combines the automation of ORM systems with flexibility and performance of hand-coded persistent layer.

Notice that the C-Gen system encapsulates many features common to the database access code. The generated code retrieves connection strings from application configuration files, establishes connection to the database, sends and retrieves data, manages transactions, etc. Of particular importance is built-in validation that prevents SQL injections and similar types of attacks, making the generated code more secure by default. All these features are provided without any efforts from application developers. However, if the requirements of particular applications conflict with the default implementation, it can be easily changed by editing templates. This is another advantage of the C-Gen system compared to ORM solutions.

In some cases there is a need to customize the generation process. The C-Gen system provides a variety of options aimed at customization. The system provides capabilities to choose which code is generated (partial generation) and customize business objects. Using partial generation prevents the system from generating an unused code. It allows the developer to customize the generation process by specifying which features should be generated.

Another customization option affects generated classes representing entities from the domain model. In some cases there is a need to extend generated classes, e.g. add some properties that are not saved to database, provide additional methods or override generated methods. In order to support such scenarios, the C-Gen generates two classes per entity: abstract and final sealed class. The abstract class contains automatically generated implementation of all the methods. Final sealed class inherits from an abstract class. Therefore it is possible to override some method in final sealed class, implement some interface, define custom logic etc. Final sealed class will not be overwritten during the next code generation.

4 Performance Evaluation

To evaluate advantages of our approach, we have compared the performance of the persistence layer generated by the C-Gen system with a code generated by NetTiers [7]. For an 18Mb sample database (representing an Internet shop) we have generated all relevant code using both systems. Then we measured the performance of *Select* and *Insert* operations for different query loads. Measurements were performed on a server with Core i5-2500k 3.3 GHz CPU and 8Gb RAM, running Windows 7 x64 SP1 and Microsoft SQL Server 2008 R2 x64 Express. The results of performance measurements are shown on Fig. 1.

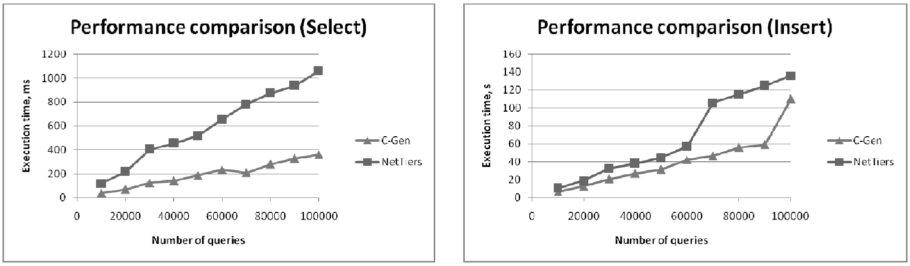


Fig. 1. Performance comparison of C-Gen and NetTiers systems

As can be seen from measurement results, the C-Gen system generates more efficient code: for *Select* operations it is about 3 times faster compared to NetTiers, and for *Insert* operations – 1.5-2 times faster. Another advantage of the C-Gen system is much smaller size of the generated code: all generated code for C-Gen is 1.2Mb, compared to 13.3Mb for NetTiers.

5 Conclusion

In this paper we have described our approach to creating a persistence layer to connect object-oriented and relational data. Our approach focuses on providing high-performance and flexible solution with high degree of automation. To this end we use a text template system to generate SQL queries, business classes and data access methods. We have implemented the C-Gen code generation system that creates easy to use and efficient persistence layer based on the database structure. Code generation mechanisms based on T4 text templates allow to generate the source code in any programming language. As a result the developer can focus on implementing more complex logic in custom stored procedures. Also the overall performance of application is improved. The system has built-in data validation mechanisms that prevent SQL injections and similar attacks. The application developer can interact with the database through strongly-typed objects that avoid runtime errors. The system is easily extensible and customizable by modifying templates, writing custom stored procedures, removing unneeded features and extending generated classes. Performance evaluation demonstrates high efficiency of the code generated by the C-Gen system compared to the similar system (NetTiers), both in terms of code size (10 times smaller) and execution speed (1.5 – 3 times faster).

Further research directions include the development of templates to automate other routine development tasks, such as the creation of user interface components for data editing. Another possible direction is extraction of templates from existing code to simplify the automation of development process and capture domain knowledge.

References

1. Bauer, C., King, G.: Java Persistence with Hibernate. Manning, New York (2007)
2. Booch, G., Maksimchuk, R.A., Engle, M.W.: Object-Oriented Analysis and Design with Applications. Addison-Wesley (2007)
3. Coad, P., Nicola, J.: Object-Oriented Programming. Prentice Hall, Upper Sadder River (1993)
4. Code generation and T4 Text Template, http://msdn.microsoft.com/en-en/library/bb126445.aspx
5. CodeSmith tools, http://www.codesmithtools.com/
6. Kirsten, W., Ihringer, M., Rudd, A.: Object-Oriented Application Development Using the Cache Postrelational Database, 2nd edn. Springer (2003)
7. .netTiers Application Framework, http://nettiers.com/
8. O'Neil, E.J.: 2008. Object/relational mapping 2008: hibernate and the entity data model (edm). In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), pp. 1351–1356 (2008)
9. Russell, C.: Bridging the Object-Relational Divide. Queue 6(3), 18–28 (2008)
10. Sanderson, D.: Programming Google App Engine: Build and Run Scalable Web Apps on Google's Infrastructure, 1st edn. O'Reilly (2009)
11. Tamer A-zsu, M., Valduriez, P.: Principles of Distributed Database Systems, 3rd edn. Springer (2011)

Are Anti-goals Needed in Systems Design?

Roland Kaschek

The University of the Faroe Islands, Department of Natural Sciences (NVD)
Noatun 3, 100 Torshavn, The Faroe Islands
rolandkaschek@gmail.com

Abstract. I report on a study into enhancing communication privacy in virtual worlds. That study used steganography as guiding idea. Its approach could easily be adapted to work for further virtual worlds. Liability and cost issues, however, might make the related providers oppose to inhabitants of their world practice it. I try to conceptualize the underlying problem in terms of systems development. Then I raise the question if these aspects of computerized systems need to be considered more careful and to be addressed by software processes.

Keywords: goal, anti-goal, steganography, communication, privacy.

1 Introduction

Information is a key asset in modern society. Even new forms of crime, such as identity theft, have emerged in response. Modern information technology, moreover, is used for "steeling" information such as product or production parameters, passwords for Internet banking and similar, [9]. Personal information may need protection too so as to make mobbing less likely and effective or to control one's perception in a community. Some users also might want to prevent parts of their online life to be the target of marketing activities. Many sorts of activities rely on the possibility of private communication. In [1] even intelligence agents have been used as an example of actors who rely on it. Freely following Bateson one can define information as such records that can be used to make a difference. Access to information must then be restricted when and if unacceptable consequences may result from that information being used the wrong way or by the wrong people. Which information to subject to access restrictions is, moreover, a personal decision. Therefore additionally to the reasons indicated there may be many more reasons to wish for an enhanced information privacy.

To be on the safe side information privacy must be understood as a limited and temporary one. The cost for uncovering a secret, i.e., an information with access control put on it, can be assessed as the sum of the related procuring and deciphering cost. Steganography and cryptography can aid in increasing the related cost, respectively. Some texts, such as the Voynich manuscript, [13], could not be deciphered yet. A strong information privacy requirement, however, forces to assume that after the discovery of an ongoing communication, procuring and deciphering the

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 185–190, 2013.

message is merely a matter of time, resources and scruple. Cryptography has inherent principle limitations as it disguises the message but does not hide communication.

Enhancing communication privacy based on steganography, i.e., hidden communication, additionally to the interlocutors involves the unwilling provider of the required communication infrastructure as well as the fellow users of the “abused” virtual worlds. The provider of an “abused” virtual world might dislike such hidden communication as it might involve legal issues and deteriorate the usage experience of those not involved in that communication. Therefore hidden communication affects the information systems community.

In the struggle concerning the protection of intellectual property it plays a role whether or not the provider of a web site should be held responsible for the content or activity on pages to which they link to. If such a responsibility would be confirmed then one could argue that virtual-world-providers are liable for the doings of inhabitants of the provided virtual worlds. That responsibility could be effective regardless of the purpose of such virtual worlds and it might be considered as immaterial whether the provider did explicitly encourage hidden communication.

System design usually follows a problem solving approach. In it the term problem means a negatively assessed difference between an as-is and a to-be state. Accordingly a stimulus-state-response medium first is conceived of as a means of solving the problem at hand. That medium is then designed and implemented following the “system” metaphor. The potential user behavior in system development stages therefore is mainly considered from the view of intended system functionality. Concepts and techniques thus are not in the scope of the system developers that would allow them to address the user activity pursuing a goal the system under development is not supposed to achieve.

Traditionally in requirements engineering goals of computerized systems have been addressed in terms of “functional” vs. “non-functional” requirements. The former and the latter can be understood as requirement to provide system functionality and system quality, respectively. System quality refers to system states and transitions among these. Call an anti-goal of a system a state that system has to stay clear of and the transition into is not the consequence of executing any individual particular set of system functionalities. Then hidden communication is an anti-goal. It cannot simply be ignored since it clearly has strong evil potential. It is, difficult to do something about it since it is designed to go undetected.

Should anti-goals be incorporated into software processes? Right now I cannot answer that. There is, however, evidence that points at a need of using anti-goals. For example, the related term anti-pattern is now established in software design. Also human society controls human behavior by goals and by anti-goals such as commandments and bans, respectively. We moreover do not only know of rewards also punishment is used for educational purposes. While in a closed world a solely reward based approach could work I don't think the same goes for an open system. Perhaps computing already has penetrated society to such an extent and quality that computerized systems such as virtual worlds should be considered as open rather than closed systems. Web systems, in a different context, indeed have been characterized as open systems, [11]. Making new technology available encourages new behaviors to emerge and old behaviors to persist. Not always do inventors know in advance which effects on human behaviors they are going to trigger. Together with society standards his implies that constraints may be important that cannot be addressed from solely within new technology.

2 Basic Terminology and Related Work

A **virtual world** is a technology backed media for a community to pursue its lifestyle. It provides a peculiar usage experience and enables recording, storing, and disseminating linguistic expressions, as well as drawing related inferences. Thus, [8] p. 11, according to Langefors it is an information system.

Encryption is the process of replacing a **message** by a character sequence, the **cipher**, such that for an insider and an outsider the cost of message reconstruction, i.e., **deciphering**, is acceptable and prohibitively high, respectively. Of course any assessment of the deciphering cost relies on knowledge that is hard to come by and thus might be flawed, unreliable or outdated. To increase the message procurement cost is independent from encryption and may be practiced on top of it.

Steganography [10, 1] is the discipline of hiding data in other data. This paper addresses the use of steganography within virtual worlds. Its focus is hiding communication in ongoing apparent communication. To hide communication requires to hide the messages and the conversation context. Message hiding may be accomplished by chopping it to pieces and to generate for each piece a symbol sequence the **carrier sentence** that would be most suitable for secretly transporting that message piece as a so-called **payload**. To hide the context of a conversation may be accomplished by practicing non-suspicious behavior that, however, to the insider is a signal. Such approach likely is less vulnerable to **steganalysis** (detecting and if possible recovering steganographic content) than the approaches in [4, 5, 3, 7] because these put the payload into given file types that easily can be scanned and the outcome matched with word distribution statistics of the suspected language.

The papers [12, 2, 6] discuss hiding conversation in conversation rather than in conversation meta data. The first of these papers, however, suggests using TCP / IP as stego media. It obviously thus is not suited for daily use by ordinary people. The second makes assumptions that I consider as too strict, i.e., exactly two agents wish to communicate and are aware of being under surveillance by exactly one agent. The third of the aforementioned papers uses an extension of iChat, i.e., Apple's video chat. It appears to be too simple to result in an effective information hiding approach.

3 The Experiments

The experiments[1] this paper draws from had been briefly planned and were based on communication protocols. The latter were supposed to guide interlocutors and thought to

[1] The experimenters were A. Akhmetov, R. Arslanov, U. Nielsen, O. Prokudin and D. Yan. A. Tulegenova contributed to an initial paper version. Back in 2008 they were my students at the KIMEP´s Bang COB.

have the key dimensions: (1) actor, i.e., their number and role; (2) sign system; (3) communication mode, i.e., monolog, dialog or conversation; (4) meta level, i.e., object vs. meta language; (5) communication channel, (6) media type; and (7) parameters, i.e., their number and kind, i.e., variable vs. constant and explicit vs. implicit.

The experiments took place at about 8 pm of Almaty local time on December 4th. 2008. The virtual worlds were **WoW** (World of Warcraft), **RO** (Ragnarok), **PWR** (Perfect World Russia), **VRU** (Vkontakte.ru) and **FB** (Facebook). In each case the experiment had been announced to openly transmit a message and the world inhabitants had been asked to recover it. Immediately before experiment begin for each virtual world I generated a message and issued it to the related experimenter. My English messages ad-hoc had to be translated to Russian since that was the language used in most of the virtual worlds.

In **WoW** the message was: “Merchants have crossed the ocean in search for love!” It was transformed to: (1) “Merchants”, (2) “have crossed”, (3) “ocean”, (4) “in”, (5) “search”, and (6) “for love”. Prior to experiment begin the participants were introduced into their protocol. A reward of USD 5 was promised for figuring out the message within two days. The payload, however, was transmitted correctly and nobody claimed the reward.

In **RO e**ighteen persons were involved in transmitting the message. They virtually met at the game's marketplace. The message was “We come from far and have to go a long way”. It was transformed to: (1) “We”, (2) “come”, (3) “from far”, (4) “and”, (5) “have to”, (6) “go”, (7) “a long”, and (8) “way”. The sender then opened a chat room with the title: “S>8 red ribbons”. Then he made his avatar sit down for five minutes so everyone would notice. He waived a flag as sign of transmission being started.

On reception of the carrier sentences the intermediaries recovered the payload words and transmitted them to the receiver. Both sets of used intermediaries sent the same payload and thus no error was signaled by the receiver. The reward set out for figuring out the message was a very rare and very valuable one in RO. The experimenter received more than 10 related responses. The payload was successfully transmitted, no errors occurred and nobody figured out the message.

In **PWR** the message was “Walking tall is not a sport. It is a passion!”. It was transformed to: (1) “walking”, (2) “with sense”, (3) “of self-respect”, (4) “isn’t sport”, (5) “it’s”, and (6) “lifestyle!”.

The intermediary provided carrier phrases have been transmitted via PWR-global chat. A reward was promised for pointing out any suspicious words and one for recovering the message, respectively. The payload was received OK. Two inhabitants earned the suspicious-words-award. But nobody earned the main price!

In **VRU** the experiment took place in a group on the game “Mafia”. Breaking the protocol and obstructing transmission was the challenge for the VRU public. No reward was, however, promised. The time given was 48 hours. The setup has been explained to sender, receiver and the intermediaries in advance. The latter additionally were trained in carrier sentence selection. That was difficult and time consuming. Due to the experiment time circumstances the “wall” was used for communication.

The message was: “The bear is touching the stars with his tail”. It had been transformed to: (1) “Bear”, (2) “is touching”, (3) “the stars”, (4) “with his”, and (5) “tail”. The transmission started with the sender commenting on users' photos. To create confusion the non-payload words “Game” and “Time” had been added.

The payload finally was transmitted and five VRU inhabitants tried and failed to break the protocol. A co-experimenter identified the word “bear” as suspicious.

In **FB** the message was: “Ain’t no sunshine when she’s gone. It’s not warm when she’s away”. One time repeated selection modulo three in the message was used to identify the payload words that had to be transmitted via either of the three channels: group, image, and wall.

Initially the receiver had to count the number of pictures uploaded at the experiment's day on the sender’s page to get the number of payload words. Then he had to attend three specified groups and look at the messages sent by the sender. After that the receiver had to visit Peteris Ciematnieks’s personal page to pick up a sender comment made to a specific picture. Thereafter, he had to go to Charles Hanson’s personal page and have a look at the sender's posting to the “wall”. Finally the receiver had to recover the message by protocol. Any transmission failure would have to be signaled by changing the profile picture. The message, however, was transmitted correctly.

4 Discussion

The above experiments suggest that procuring a message may not be easy when suitable steganographic protocols are employed for carrier sentence creation. The related protocols currently are in a proof of concept status and not ready yet for practical use. They seem in particular not to be suited for transmitting long messages.

Mixing signal systems such as activity patterns, colors, letters, character sets and purposely making mistakes seem to increase the procuring and deciphering cost of transmitted messages. However, choosing from the established behaviors those that effectively aid hidden communication remains a difficult issue.

It seems that adding distracting content or surplus activity may contribute to the strength of a steganographic communication protocols. This may even be more so if the protocol guides in hiding communication context in several ongoing conversations. Using intermediaries enables transmitting payload in parallel maybe even in a variety of virtual worlds using different intermediary protocols.

Steganographic communication protocols have potentials not yet addressed in the experiments above. For example various unrelated messages might be transmitted as payload. These could correspond to different levels of protocol breaking and thus introduce deception into the payload. Someone breaking the protocol only partially not necessarily would notice that. More complex protocols might require IT support.

For systems design to be better prepared for dealing with anti-goals a list of anti-goals is needed. Once such anti-goals have been defined one could try to develop operation surveillance and log analysis techniques that are capable of uncovering hidden communication. These, of course, should be combined with a fair use approach. Finally self-healing capabilities should be developed.

Putting to use new technology may have the desired consequences but often also will have undesired and unanticipated ones. Some of the latter result from legal public-space-behavior restrictions put on system providers or human beings in general. Unanticipated consequences of new technology may also result from it doing that good a job that new capabilities emerge. The unanticipated consequences thus may not be tightly connected to any particular set of system functionalites or qualities.

To deal with these in systems development and operation the term anti-goal has been coined in this paper. Hidden communication has been discussed briefly as an example anti-goal. A case study into steganographic enhancement of communication privacy has been used for that. For a variety of reasons it is plausible to expect that some virtual world inhabitants are not satisfied with the communication privacy provided these days. Therefore it is likely that techniques like the ones briefly described in this paper are being tried out and developed right now. Maybe the best strategy to fight such developments is prevention. It could take several forms such as by law enhance privacy, and introduce the capability to disable tracking, advertizing and analyzing user behavior. Maybe inhabitants of virtual worlds should be given rights by law the violation of which could be taken to court and tried. The information systems community should, in my view, seriously address problems related to anti-goals.

References

1. Artz, D.: Digital steganography: hiding data within data. IEEE Internet Computing, p. 75 (May-June 2001)
2. Bergmair, R., Katzenbeisser, S.: Content-aware steganography: about lazy prisoners and narrow-minded wardens. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 109–123. Springer, Heidelberg (2007)
3. Chen, M., Zhang, R., Niu, X., Yang, Y.: Analysis of current steganography tools: classification & features. In: Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2006), doi:0-7695-2745-0/06
4. Chen, Z., Huang, L., Yu, Z., Zhao, X., Zheng, X.: Effective linguistic steganography detection. In: IEEE 8th International Conference on Computer and Information Technology Workshops (2008), doi:10.1109/CIT
5. Chen, Z., Huang, L., Yu, Z., Li, L., Yang, W.: A Statistical Algorithm for Linguistic Steganography Detection Based on Distribution of Words. In: The Third International Conference on Availability, Reliability and Security (2008), doi:10.1109/ARES.2008.61
6. Craver, S., Li, E., Yu, J., Atakli, I.: A supraliminal channel in a videoconferencing application. In: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 283–293. Springer, Heidelberg (2008)
7. CSL: CyberScience Laboratory functional analysis of StegAlyzer ASTM Version 3.0, `http://www.cybersciencelab.com` (accessed on April 3, 2009)
8. Hirschheim, R., Klein, H., Lyytinen, K.: Information systems development and data modeling, conceptual and philosophical foundations. Cambridge University Press (1995)
9. IC3:. 2008 Internet crime report. US Internet Crime Complaint Center (2008), `http://www.nw3c.org/`
10. Johnson, N., Jajodia, S.: Exploring steganography: seeing the unseen, pp. 26–34. IEEE Computer (February 1998)
11. Kaschek, R., Schewe, K.-D., Wallace, C., Matthews, C.: Story boarding for Web-based information systems. In: Taniar, D., Wenny Rahayu, J. (eds.) Web Information Systems, ch. I, pp. 1–33. IDEA Group Publishing Inc., Hershey (2004)
12. Lucena, N.B., Pease, J., Yadollahpour, P., Chapin, S.J.: Syntax and semantics-preserving application-layer protocol steganography. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 164–179. Springer, Heidelberg (2004)
13. Schmeh, K.: Das rätselhafteste Buch der Welt. Telepolis (February 19, 2012)

Model and Tools for Multi-dimensional Approach to Requirements Behavior Analysis

Mykola Tkachuk and Irina Martinkus

Computed-Aided Management Systems Department,
National Technical University "Kharkiv Polytechnic Institute",
Frunze str., 21, Kharkiv, Ukraine
tka@kpi.kharkov.ua, imartinkus@gmail.com

Abstract. Requirements modeling and analysis (RMA) are important and weak-formalized activities in the whole requirements engineering framework. In this paper an advanced approach to RMA is proposed, which provides multi-dimensional estimation of requirements in appropriate attribute spaces, with respect to their both static features and dynamic behavior. The given test-case shows a complex requirements behavior and visualizes requirements modeling process, in order to support decision making in software development.

Keywords: requirement modeling, dynamic behavior, linguistic variable, attribute space.

1 Introduction: Research Actuality and Aims

As it is well-known, requirements modeling and analysis (RMA) are ones of the most important and weak-formalized stages in software development process [1,2]. Among existing approaches to RMA just few of them take into account not only the static characteristics of requirements (see, e.g. in [3]), but also their dynamics features: e.g., dynamic requirements adaptation in [4] or simulation requirements evolution in [5] and some others. From the other hand, many authors are using the term "dynamic process" to describe a requirements engineering framework, but actually do not reflect the dynamic nature of requirements behavior itself (e.g., see in [6]). These aspects of RMA are especially critical for support and maintenance of legacy software systems (LSS), because these activities take about 50-55% of all projects' costs in any software life-cycle model [7]. In order to eliminate this gap in the whole requirements engineering framework we propose to elaborate a multi-dimensional approach to RMA, which takes into account a requirements functional complexity and their dynamic features as well. In Section 2 we introduce some terms and definitions needed in our approach, and propose the 2-spaces metaphor for our RMA-approach, Section 3 presents the mapping procedure between these attribute spaces, and in Section 4 we briefly discuss implementation issues and test-case results. Finally, conclusions and future work are addressed in Section 5.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 191–198, 2013.

2 Some Terms and Definitions

The proposed multi-dimensional approach to RMA is based on the following terms and definitions, and utilizes the space-focused metaphor introduced below.

Def#1. *System Type* (ST) is an integrated characteristic of any LSS given as tuple:

$$ST =< Structural\ Complexity, Requiremen\ tRank > \tag{1}$$

The first parameter estimates a complexity level of given LSS, and the second one represents status of its requirements: their both static features and dynamic behavior. A software structural complexity can be defined using the appropriate metrics (see e.g., in [8]), and for this purpose the following collection of such metrics was choused: *Cyclomatic Complexity* (V) *Weighted Method Complexity* (WMC), *Lack of Cohesion Methods* (LCOM), *Coupling Between Objects* (CBO), *Response For Class* (RFC), *Instability* (I), *Abstractness* (A), *Distance from main sequence* (D). All these values can be calculated using standard software tools, e. g. *EclipseMetricPlugin* for the Eclipse platform. To define final structural complexity level we need the weighted coefficients for each metric, which were calculated with help of Analytic Hierarchy Process (AHP) method [9]. In this way we have obtained the integrated formula for LSS structural complexity (SC) estimation (2):

$$\begin{aligned} SC = K_V avgV + K_{WMC} avgWMC + K_{LCOM} avgLCOM + K_{CBO} avgCBO + \\ + K_{RFC} avgRFC + K_I avgI + K_A avgA + K_D avgD \end{aligned} \tag{2}$$

To evaluate the final structural complexity of given LSS we have elaborated the following scale (3):

$$\begin{gathered} SC_{Min} \le Low < \frac{2 * SC_{Min} + SC_{Max}}{3} \\ \frac{2 * SC_{Min} + SC_{Max}}{3} \le Medium \le \frac{SC_{Min} + 2 * SC_{Max}}{3} \\ \frac{SC_{Min} + 2 * SC_{Max}}{3} < High \le SC_{Max} \end{gathered} \tag{3}$$

In order to define correctly the second parameter given in formula (1), we propose to consider two relevant requirement features: grade of it's *Priority* and level of *Complexity*.

Def#2. *Requirements Rank* is a qualitative characteristic of LSS defined as tuple:

$$RequirementRank =< Priority, Complexty > \tag{4}$$

In the modern requirement management systems like IBM Rational Requisite Pro [10], CalibreRM [11] and some others, requirements *Priority* and *Complexity* are usually characterized by experts in informal way, e.g. using such terms as: *"Low",* *"Medium", "High",* etc. In more correct form it could be done basing on the notion of linguistic variable [12].

Def#3. *Linguistic variable* (LV) is a tuple L which includes the following items:

$$L =< X, T(X), U, G, M > \tag{5}$$

where X is a name of LV; $T(X)$ is a term-set of X, U is an universal set of numeric values; G is a syntax rule, which generates term-set $T(X)$; M is a semantic rule, which maps to each item of X it's target value $M(X)$, and $M(X)$ designates a fuzzy subset of U. Any LV-value from X is characterized by a membership function $c : U \to [0,1]$, e.g. as a membership function the Harrington's function can be used [13], namely:

$$d = e^{-e^{-R}} \tag{6}$$

where d is a value of membership function's scale; R is a value of a linguistic scale of L. The inflection points are: $d(0) = \frac{1}{e} \approx 0.37$ and $d(0.78) = 1 - \frac{1}{e} \approx 0.63$.

There are also different approaches to estimate requirements complexity in qualitative form, e.g.: *Functional Size Measurement* (FSM) represented in ISO/IEC standard [14], *Unadjusted Use Case Points* (UNC), based on use-case analysis [15], *Early Function Points Analysis* (EFPA) and some others. Exactly the EFPA-method is used in our approach, because it can be applied on early software development phases to provide a correct estimation of requirements complexity level in target system.

Taking into account definition (5), we can construct the appropriate LVs for requirements *Priority* and *Complexity* respectively as following:

$$\begin{aligned} &X : Priority \\ &T(Priority) = \{"neutral", "actual", "immidiate"\} \end{aligned} \tag{7}$$

U – a universal set, for membership function an encoded scale is used, w.r.t. experts' opinion (-6;-4;-2;0;2;4;6), see in [13] for more details;

$$\begin{aligned} &X : Complexity \\ &T(Complexity) = \{"low", "medium", "high"\} \end{aligned} \tag{8}$$

U – is an EFPA universal set (defined after applying EFPA methods to estimating initial requirements);

M – the Harrington's membership function (6).

Basing on definitions introduced above, we can formulate the following task:

to elaborate a mapping procedure between 2 *attribute spaces*, which are defined with following LVs, namely

- space *"Requirements Rank":* it has the axes *"Priority"* and *" Complexity"*;
- space "*System Type*": it has axes "*Requirements Rank*" and "*Structural Complexity*".

The next Section 3 represents the appropriate solution for mapping procedure between these attribute spaces.

3 Mapping Procedure between Attribute Spaces

To connect two attribute spaces proposed above (see Section 2) their shared axis: characteristic *"Requirements Rank"* defined in (4) has to be used. This mapping can be provided using the following procedure:

a. Break the attribute space *"Requirements Rank"* **into 3 sub-areas**: to use for this purpose k-means clustering method [16], which allows to get the required, previously known number of clusters, in any initial dimension of the space to have a 3 levels of requirements rank *{"low", "medium", "high"}*.

b. Determine weigh of each cluster: we have space with dimensions *3x3*, let's fix that:

- simple cluster w_1=1;
- average cluster w_2=2;
- complex cluster w_3=3.

Than the AHP method can be applied, in this way we get normalized weights for each cluster, namely:

- simple cluster with W_1=0.17;
- average cluster with W_2=0.33;
- complex cluster with W_3=0.5.

c. Calculation of requirement's rank values: it can be provided as

$$Rank = \frac{\sum_{i=1}^{n} W_i (\# R)_i}{n} \tag{9}$$

where W_i is a weight of i-th cluster; n is a total number of requirements; $\# R_i$ – number of requirements placed in i-th cluster;

d. Determination of scale which *Requirement Rank* will be defined with:

1. To determine the boundary points of the scale: lower threshold and upper rank threshold rank. They will be equal to corresponds weights of clusters : $Rank_d = \mathrm{W1} = 0.17$; upper threshold value of rank will be $Rank_u = \mathrm{W2} = 0.33$; the average rank will be $Rank_m = \mathrm{W3} = 0.5$.
2. Determine the intervals corresponding to the value axis "Rank requirements" {low, medium, high}, in the space "type system." To do this, there are already thresholds rank requirements, above and below which $Rank$ can not be. It's needed to defined μ–neighborhood ($\mu = middle$) of middle rank $Rank_m$

$$O_\mu(Rank_m) = \{x : | x - Rank_m | < \mu\}$$

3. To determining low limit of μ – neighborhood let's assume that 1 objects we have in simple cluster and 1 in average
$$Rank_{\mu 1} = \frac{W1 + W2}{2} = \frac{0.17 + 0.33}{2} = 0.25$$
4. To determining upper limit let's assume that we have 1 average cluster and 1 in complex
$$Rank_{\mu 2} = \frac{W2 + W3}{2} = \frac{0.33 + 0.5}{2} = 0.415 ,$$
5. μ – neighborhood of average rank $Rank_m = 0.33$ will be 0,0825:
$$O_\mu(Rank_m) = \{x : |x - Rank_m| < 0.0825\}$$
6. Finally, we get the following scale:

$$Rank = \begin{cases} [Rank_d; Rank_{\mu 1}) \\ [Rank_{\mu 1}; Rank_{\mu 2}) \\ [Rank_{\mu 2}; Rank_u] \end{cases} \quad \text{i.g.} \quad Rank = \begin{cases} [0.17;0.25) \\ [0.25;0.415) \\ [0.415;0.5] \end{cases}$$

When we have got final values of *Structural Complexity* and *Requirement Rank*, we can define an appropriate position for estimated LSS in the attribute space "*System Type*" (see Fig. 3 in Section 4).

4 Implementation, Test-Case and Results Analyzing

The approach introduced above currently is realized in the prospective CASE - tool. The common functionality of this tool is shown in Fig. 1.

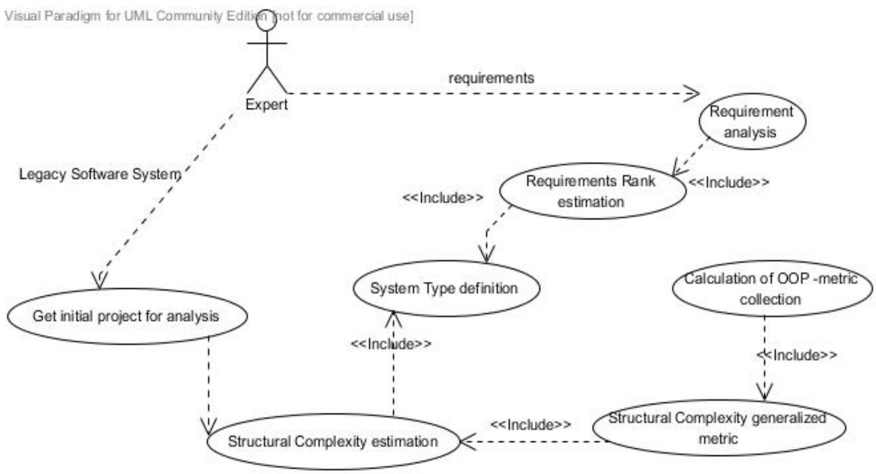
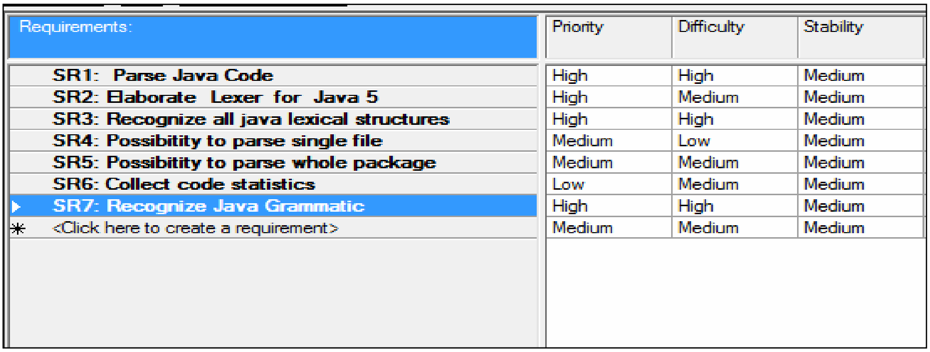


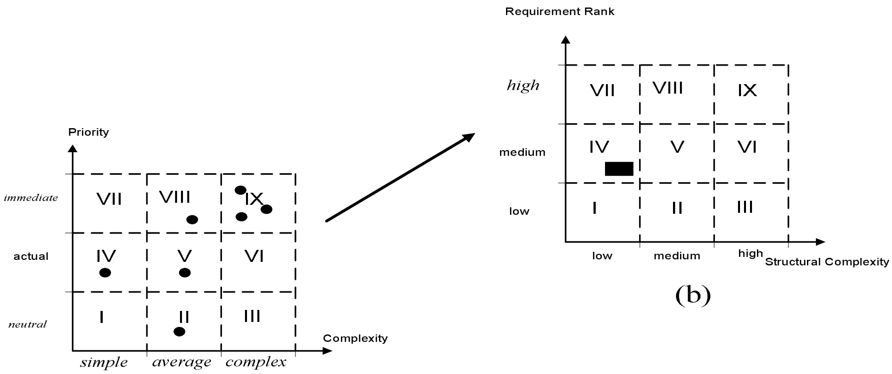
Fig. 1. The approach to "*System Type*" definition: Use-Case Diagram

Let we have for test-case the demo-software application “Java Parser”, and assume there is the appropriate list of its requirements collected in the RMS RationalRequisitePro [10]. The example of its user’s interface is presented in Fig.2. Taking into account their features “*Priority*” and “*Complexity*” (or attribute “Difficulty” in terms of RMS), and using formulas (7) and (8) to define values of appropriate linguistic variables, we have to allocate the given requirements in the space “*Requirements Rank*” as it is shown in Fig.3, (a).



Requirements:	Priority	Difficulty	Stability
SR1: Parse Java Code	High	High	Medium
SR2: Elaborate Lexer for Java 5	High	Medium	Medium
SR3: Recognize all java lexical structures	High	High	Medium
SR4: Possibitity to parse single file	Medium	Low	Medium
SR5: Possibitity to parse whole package	Medium	Medium	Medium
SR6: Collect code statistics	Low	Medium	Medium
SR7: Recognize Java Grammatic	High	High	Medium
<Click here to create a requirement>	Medium	Medium	Medium

Fig. 2. The list of “Java Parser” requirements in RationalRequisitePro (fragment)



(a)

Fig. 3. (a) – the initial allocation of system’s requirements in the space *“Requirement Rank”*; (b) – the mapped system’s position in the space *“System Type”*

Using the elaborated mapping procedure (see Section 3), we have got the following value for parameter *Rank:*

Rank =(0.17*2+0.33*2+0.5*3)/7 =(0.34+0,66+1,5)/7 = 2.5/7 = 0,357,

which can be converted into the target linguistic variable’s value:

Requirement Rank = “Medium”,

and the mapped system’s position is shown in Fig. 3, (b).

This result illustrates the main advantages of the proposed approach, namely: 1) we are able to estimate current state of system requirements w.r.t. their static and dynamic features; 2) basing on this estimation, we can define an appropriate type of investigated software system (e.g., some LSS in maintenance process), taking into account it's structural complexity and dynamic requirements behavior as well.

5 Conclusions and Future Work

In this paper an advanced approach to requirements modeling and analysis is proposed, which provides multi-dimensional estimation of requirements in 2 interconnected attribute spaces, and takes into account their both static and dynamic features. The modeling components of proposed approach are elaborated using linguistic-based evaluation of requirements priority, and customizing early-function points method for requirements complexity estimation. As possible future work the following tasks have to be considered: 1) to extend the proposed approach with respect to modeling and analyzing of non-functional software requirements, 2) to implement the full-functional CASE-tool to realize this approach, and which could be integrated with already existing requirements management systems.

References

1. Wiggers, K.: Software Requirements. Microsoft Press (2003)
2. Elizabeth, M., Hull, C., Ken, J., Dick, J.: Requirements Engineering, 3rd edn., pp. I–XVIII. Springer (2010)
3. Nguyen, L., Swatman, P.A.: Managing the Requirements Engineering Rrocess. Requirements Engineering 8(1), 55–68 (2003)
4. Jureta, I.J., Faulkner, S., Thiran, P.: Dynamic Requirements Specification for Adaptable and Open Service-Oriented Systems. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 270–282. Springer, Heidelberg (2007)
5. Port, D., Olkov, A., Menzies, T.: Using Simulation to Investigate Requirements Prioritization Strategies. In: 23rd IEEE/ACM International Conference on Automated Software Engineering, ASE 2008, pp. 268–277 (2008)
6. Williams, D.: Towards a System Dynamic Theory of Requirements Engineering Process; School of Computing, Information Systems and Mathematics. In: The19th International Conference of the System Dynamics Society, pp. 1–24 (2001)
7. Software life-cycle costs, `http://www.12207.com`
8. Aggarwal, K.K.: Software Design Metrics for Object-Oriented Software. Journal of Object Technology 6(1), 121–138 (2006)
9. Saaty, T.L.: Fundamentals of the Analytic Hierarchy Process. RWS Publications (2000)
10. Rational RequisitePro, `http://www-01.ibm.com/software/awdtools/reqpro`
11. Borland CaliberRM, `http://www.borland.com/products/caliber/`
12. Zadeh, L.A.: Fuzzy Sets. WorldSciBook (1976)

13. Trautmann, H., Claus, W.: On the Distribution of the Desirability Index Using Harrington's Desirability Function. Metrika 63(2), 207–213 (2006)
14. Information Technology - Software Measurement - Functional Size Measurement. Part 1: Definition of Concepts: ISO/IEC 14143.1:1998, http://www.ieee.org/publications_standards
15. Henderson-Sellers, B., Zowghi, D., Klemola, T., Parasuram, S.: Sizing use cases: How to create a standard metrical approach. In: Bellahsène, Z., Patel, D., Rolland, C. (eds.) OOIS 2002. LNCS, vol. 2425, pp. 409–421. Springer, Heidelberg (2002)
16. MacQueen, J.B.: Some Methods for Classification and Analysis of Multi-variate Observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Berkeley (1967)

A Layered Architecture Approach for Large-Scale Data Warehouse Systems

Thorsten Winsemann[1,2], Veit Köppen[2], Andreas Lübcke[2], and Gunter Saake[2]

[1] SAP Deutschland AG & Co. KG, Großer Grasbrook 17, 22047 Hamburg, Germany
Thorsten.Winsemann@t-online.de
[2] Otto-von-Guericke-Universität, Universitätsplatz 2, 39106 Magdeburg, Germany
{Veit.Koeppen,Andreas.Luebcke,Gunter.Saake}@ovgu.de

Abstract. A proper architecture is significant to cope with complex requirements of today's large-scale Data Warehouses. We compare the assignment of so-called reference architectures with an architecture of dedicated layers to satisfactorily face those requirements. Moreover, we point out additional expenses and resulting advantages of this layered approach.

1 Introduction

We define Enterprise Data Warehouses (EDW) as large-scale Data Warehouses (DW) for supporting decision-making on all organizational levels and in all divisions. EDW are an important basis for applications such as Business Intelligence, planning, and Customer Relationship Management. As they are embedded in an enterprise-wide system landscape, EDW must provide a common view on centralized, accurate, harmonized, consistent, and integrated data. They cover all areas of an enterprise, including the collection and distribution of huge amounts of data with a multitude of heterogeneous sources. World-wide range of use means that data from different time-zones must be integrated. Frequently, 24x7-hours data availability has to be guaranteed, facing the problem of concurrent loading and querying. There are high requirements to data: ad-hoc access, near real-time availability, high quality, fine-granular levels of detail, and a long time horizon. Moreover, new or changing needs for information must be flexibly and promptly satisfied. Such complex requirements for EDW need enhancements and refinements to the architecture, compared to traditional ones, which we outline in this paper.

2 Traditional vs. Layered Architecture

In literature, DW reference architectures are build with three to five rather rough levels [1, 2, 3, 4]; Figure 1 shows a simplified model. Data are extracted from sources into the staging area and transformed to a common schema. Afterwards, they are loaded into the basis database and stored usually at the extracted level of detail. Based on this, data marts are built, i.e. redundant copies of data that are defined according to the users' requirements. The operational data store is used for special business cases. Practical experiences show that this rough architectural model is insufficient. Data

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 199–201, 2013.

access, e.g., for analyses, does not only occurs at data mart level, but also in the basis database, and data transformation is not restricted to the staging area, but takes place in all DW areas; cf. [5]. Besides, complexity of data processing is not expressed adequately – although it is a big problem when building and operating a DW [3, 4]. Complex processing and transformation requires a comprehensive classification within a suitable, layered architecture. It describes levels of data alteration; i.e., layers become more detailed, dedicated, and purposeful. An example is SAP's "Layered, Scalable Architecture" [5, 6, 7]; see Figure 1. We describe the layered approach based on this example, even though it is not limited to SAP-based systems.

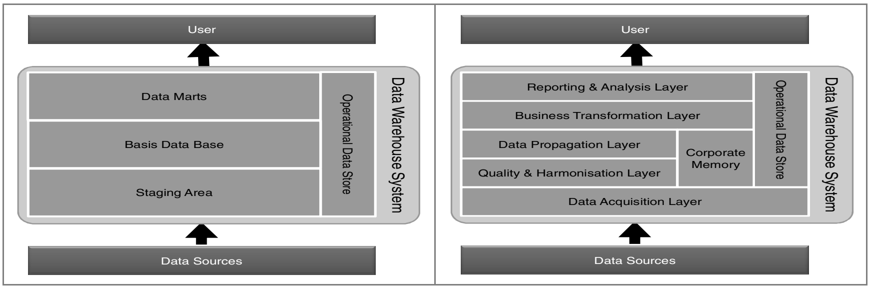


Fig. 1. Traditional DW Reference Architecture vs. Layered, Scalable Architecture

Just as the traditional approach, the layered one is a reference for designing architectures according to individual and actual requirements. However, it is more dedicated, as each layer represents a designated area, in which data are changed w.r.t. their actual format and usage. Extracted data are stored without any transformations in the *Data Acquisition Layer*. The source system is decoupled immediately and no longer strained after loading. Successfully processed data are deleted after a certain time. Yet, if they are stored in the long term, a *Corporate Memory* can be built which enables access to all loaded data. For easier reuse, administrative information can be added, e.g., data origin or timestamp. As the use is rare, storage mediums can be slower and cheaper ones. In the *Quality & Harmonisation Layer*, all operations to integrate data are made, cf. [8]. The result is harmonized and integrated data, which are stored in the *Data Propagation Layer*. It represents a single source of truth and is basis for any further data usage. Therefore, the meaning of data in here must have a clear, common, and company-wide understanding. Data are kept as granular as possible with an adequate time horizon, and the absence of business logic offers high flexibility for further data usage. In the *Business Transformation Layer*, data are transformed regarding business needs; e.g., key figures' computation or combination of data from different areas. The *Reporting & Analysis Layer* offers data "ready-to-use". Data persistence can be necessary to enhance query performance or due to complex data transformation. Yet, data can also be read from layers below. The *Operational Data Store* is mainly dedicated for special needs of data with near real-time availability.

3 Efforts and Benefits of the Layered Approach

A layered architecture initially covers complete business areas and combines data according to current needs in a subsequent step. In contrast, the traditional approach defines DW based on actual users' requirements. This leads to higher complexity and higher volume of extracted data. More conceptual work is necessary, so that duration and costs of the implementation are affected. However, we illustrate that several advantages justify a layered architecture. Transformation rules have to be changed as the transformation of loaded data was erroneous; e.g., currency conversions based on overaged rates. In traditional architectures, data are usually not available in a reusable format, and must be reloaded from sources. In our approach, the propagation layer holds data in a format that enables rebuilding into the overlying levels. Further, data have to be remodeled for new business requirements, e.g., sales regions are restructured. Again, data reloading from sources is usually the solution in traditional architecture, whereas computation can start from propagation layer in the layered approach. Business often requires data that are not included in the initial concept of DW's design. Even in case such data are part of already connected sources, they are not included in data extraction in traditional architectures. Hence, the data flow from source to DW must be enhanced and all data must be loaded again. In the layered architecture, we extract all potentially relevant data into the DW when a new source is extracted. Most likely, data are already kept in the propagation layer or the corporate memory, including previous ones, so that remodeling data flows from sources and extracting missing data is not necessary. Hereby, organizational problems are avoided, e.g., downtimes of source systems during extraction. As data are kept free of business logic up to the propagation layer, they must be loaded only once into the DW and are deployed several times. Hereby, system load, due to redundant extraction, staging, and storage of data, is avoided. Moreover, data remain in the corporate memory even when they are already deleted or changed in the source system. Defining a single source of truth of data supports a common, company-wide view on integrated and trustable data, w.r.t. data governance. Thereby, local or departmental data marts contain reliable information, too. Besides, data load and data supply for reporting is decoupled, and data from different time zones can be released simultaneously. Moreover, a layered architecture eases scalability. Due to openness, modularity, and flexibility of this concept, the system can easily be enhanced by integrating data streams or copying applications, e.g., in case a new business area is defined. Relevant data are transferred to the DW instantly as extraction is not limited to existing areas. As there is no business-related data modification up to the propagation layer, the DW model is not affected until necessary enhancements in the transformation and reporting layer. Here, initial additional expenses pay off.

References

1. Poe, V.: Building a Data Warehouse for decision support. Prentice Hall PTR (1996)
2. Muksch, H., Behme, W. (eds.): Das Data-Warehouse-Konzept. Gabler (2000)
3. Chamoni, P., Gluchowski, P. (eds.): Analytische Informationssysteme. Springer (2006)
4. Köppen, V., Saake, G., Sattler, K.-U.: Data Warehouse Technologien. Mitp (2012)
5. Winsemann, T., Köppen, V.: Persistence in Enterprise Data Warehouses (2012)
6. SAP AG: PDEBW1 – Layered, Scalable Architecture (LSA) for BW (2009)
7. Winsemann, T., Köppen, V., Saake, G.: A layered architecture for enterprise data warehouse systems. In: Bajec, M., Eder, J. (eds.) CAiSE Workshops 2012. LNBIP, vol. 112, pp. 192–199. Springer, Heidelberg (2012)
8. Leser, U., Naumann, F.: Informationsintegration. Dpunkt (2007)

Towards Medical Screening Information Technology: The Healthgrid-Based Approach

Karina Melnik, Olga Cherednichenko, and Vitaliy Glushko

National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine
karmens@km.ru, marxx75@mail.ru, vitaly.glushko@gmail.com

Abstract. The main task of medical information technologies is the processing of distributed electronic medical information. The solution of this task is based on the patients' electronic medical records. The paper is devoted to data analysis for early disease detection and defining risk groups in population. The architecture of healthgrid-based medical information system is suggested. Formalization of domain area based on screening ontology is proposed.

Keywords: screening, medical information technology, healthgrid, ontology, architecture.

1 Introduction

Nowadays information technologies (IT) have become necessary and available tools for medical research and clinical practice [1]. Medical information technologies (MIT) are the combination of IT and healthcare technologies. Today MIT are successfully used for solving of different medical problems. There are many publications dedicated to resolving of the task of diagnostics, processing of medical information; a lot of computer advising systems, image recognition systems and expert systems have been developed [e.g. 2, 3, 4].

Analyzing references about using of MIT, it can be noted that majority of recent publications are dedicated to eHealth problems [5], a lot of papers describe HelathGrid technologies [6].

Experience shows that MIT covers large set of tasks from diagnostics to healthcare management. One of the main tasks in healthcare is the early disease detection and prevention in population. MIT, used for processing of data associated with the detection of individuals having any pathology or risk factors of its development, we will refer to as Medical Screening Information Technologies (MSIT).

The goal of the given research is to increase the performance of early detection and prevention of diseases based on elaboration of MSIT and implementation of screening information system. In this work we consider issues of distributed medical electronic data processing and provide architecture of MSIS – medical screening information system - that supports the screening process. Such information system is aimed to be used by medical establishments. It also can be used by eHealth community as a Grid-service.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 202–204, 2013.

2 Data Processing for Medical Screening

The main part of healthcare system is medical establishments which provide medical and preventive aid to population. They are clinics, doctor's offices, hospitals etc. We consider a healthcare facility that is primarily devoted to the care of outpatients.

Screening procedures are performed for early detection of different diseases and propensities for some disorders in an organism which are key factors for providing timely medical or preventive aid. Screening results can be used for development of a set of measures that allows decreasing incidence of the diseases.

We have investigated functioning of a typical Ukrainian clinic and defined its formal business processes. We believe the similar activities take place in the regular medical establishment in many other countries.

Most of researchers come to the point that ontology can be helpful when there is a need to define a common vocabulary of terms for some application domain. There are ontologies in the domain of healthcare, for example SNOMED CT, UMLS [7]. However, for the problem of medical screening it makes sense to specify separate domain ontology.

First of all, medical screening relates to processing of patient's data to make a decision about level of patient's health and possible risks of disease development. We can single out two main approaches that can be used for disease diagnostics and prediction of its development. They are clinical examination and processing of patient record. Clinical examination involves diagnostic procedures including examination of patient and laboratory research.

History of patient treatment in medical facility is stored in patient medical record. We suppose that past diseases, heredity, age, living conditions and so on can give necessary information for matching with risk factors of the disease under consideration. All this information should be contained in patient medical record.

Therefore, screening ontology formalizes the knowledge about decision making processes for early disease detection and making predictions of development of certain diseases. We suggest three main ontologies – screening, patient record and disease – as the basis of MSIT.

The main problem of working with medical data is the access to medical information from different healthcare facilities. Solution of this issue is the using of HealthGrid. Grid infrastructure can be used in the following three directions [8]:

- distributed computing which is considered as a mainframe for support of medical decisions;
- distributed medical datastore which is used for storing, accessing and searching for data in distributed medical information by means of onthologies;
- collaborative grids which are the virtual environment for interaction of geographically distributed facilities.

HelathGrid concept can be used in the following two aspects: for individual healthcare and for epidemiological analysis. First approach provides an on demand access to patient's clinical information for solving ongoing problems. Second approach allows using of medical information of different sections of population for finding dependencies between data, risk hazards, symptoms and diseases.

3 Conclusions

Features of HealthGrid are implemented in Open Grid Services Architecture (OGSA) by means of Grid technologies and Web resources [8]. To resolve screening tasks it is proposed to use Healthgrid-based architectured. The proposed architecture consists of the following levels: Grid Control; Application Server Grid; Database Grid; Storage Grid. Storage Grid is the resource level. It includes medical information warehouses which contain results of medical researches, collected information on medical treatments, information about patients and their clinical courses. Database Grid represents the level of databases of medical establishments, laboratories, R&D centers. The Application Server Grid level contains a set of user applications for resolving required tasks. Execution of interaction protocols, authentication, security policy, administrating and setting of components of the HealthGrid network are performed on the Grid Control level. This level is responsible for control and interaction of software and hardware components of HealthGrid architecture: database management servers, application servers, storages, network components, data flows.

Thus, the amount of medical information that can be used for screening activities is increasing from day to day. The computational capabilities of HealthGrid technology should be used for its most effective processing. MSIT are promising and up to date direction of research in healthcare system at the moment. Developments in this direction will lead to increasing of the quality and speed of screening data processing that, in turn, will improve making of medical decisions.

References

1. Bemmel, J.H., Musen, M.A.: Handbook of medical informatics, Houten. Springer (1997)
2. Neural Network Applications in Medical Research, `http://www.nd.com/apps/medical.html`
3. Kulikowski, C.A.: Artificial intelligence methods and systems for medical consultation. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2(5), 464–476 (1980)
4. Stansfield, S.A.: ANGY: a rule-based expert system for identifying and isolating coronary vessels in digital angiograms. Technical Report, Proceedings of the First Conference on Artificial Intelligence Applications. IEEE Computer Society (1984)
5. Implementing e-Health in Developing Countries: Guidance and Principles. In: International Telecommunication Union (2008), `http://www.itu.int/ITU-D/cyb/app/docs/e-Health_prefinal_15092008.PDF`
6. Breton, V., Dean, K., Solomides, T.: The Healthgrid White Paper, `http://www.healthgrid.org/documents/pdf/HealthGrid-whitepaper-full.pdf`
7. International Health Terminology Standards Development organization, `http://www.ihtsdo.org/snomed-ct/`
8. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: The Physiology of the Grid. An Open Grid Services Architecture for Distributed Systems Integration, `http://www.cs.indiana.edu/~aslom/papers/ogsa_physiology_2002-02-17.pdf`

Knowledge-Oriented Approach to Requirements Engineering in Ambient-Assisted Living Domain

Volodymyr Bolshutkin[1], Claudia Steinberger[2], and Mykola Tkachuk[1]

[1] National Technical University "Kharkiv Polytechnical Institute", Kharkiv, Ukraine
vladimir.bolshutkin@gmail.com, tka@kpi.kharkov.ua
[2] Alpen-Adria-Universität, Klagenfurt, Austria
claudia.steinberger@aau.at

Abstract. This paper sketches HBMS(Human Behavior Monitoring and Support), an Ambient Assisted Living (AAL) approach deriving cognitive behavioral models from individual behavioral processes and using this knowledge "learned" to compensate individual memory gaps. In the context of HBMS individual behavioral processes represent the requirements to customize an individual assistive system.

Keywords: ambient-assisted living, requirements engineering, knowledge, behavior modeling.

In the context of demographic aging in the western world improving the quality of life for disabled and elderly people is an essential task for our society. Facing these challenges technological innovations can enhance the quality of life of older and impaired people and contribute to independent living and quality of life.

Ambient Assisted Living (AAL) solutions are developed to help elderly to live longer at the place they like most enhancing their safety and security, giving them assistance to carry out daily activities, monitoring their activities and health, getting access to emergency systems and facilitating social contacts [1,2].

So far little attention has been paid in AAL research to give persons aid to memory to carry out their daily activities although memory gaps are very typical to evolve in an advanced age. Especially elderly people often need support carrying out activities such as using technical devices(e.g. washing machine, TV-set), dealing with administrative duties, using electronic banking tools, using online-shops or simply to remember all steps of their daily life activities. Existing assistive technology systems for cognition are forcing compliance with standardized processes defined by third parties [3] and neglect established user habits. Thus the user acceptance level of such systems is often rather low. The possibility to use established cognitive behavioral processes as requirements to assist the individual later on would improve user acceptance.

Human Behavior Monitoring and Support (HBMS) [4,5] is an approach to derive cognitive behavioral models from individual behavioral processes. Knowledge "learned" in this way is stored in an "artificial memory" and can be recalled later in time to compensate gaps in the episodic memory of the respective person and to technically assist his or her activities.

H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 205–207, 2013.

Over time a person takes two different roles in HBMS: *knowledge holder* and *knowledge user*. As a knowledge holder the person affects how the knowledge user will be supported later in time. This means that the knowledge holder acts as a stakeholder and his or her knowledge acts as a set of requirements customizing HBMS for the knowledge user.

Hence requirements engineering in HBMS has some characteristics, which lead to differences compared to typical requirements engineering activities [6]:

1. *Requirements Elicitation or Gathering*: traditionally requirements are to be discovered from different system stakeholders. HBMS is supposed to support one individual using his or her own knowledge as set of requirements elicited to customize the assistive system; hence in the HBMS requirement engineering process there is no need for dealing with different knowledge holders. Currently requirements elicitation in HBMS is done observing person's behavior by a psychologist, who writes a textual description of the monitored behavior of the person. In future automated observation techniques are planned to support individual behavioral process elicitation.
2. *Requirements Analysis and Negotiation*: this activities usually include checking requirements and resolving stakeholder conflicts. Hence in the HBMS requirement engineering process there is no need for negotiations between different knowledge holders. But as several behavioral processes can lead to the same result HBMS has to handle behavior process integration and inconsistencies.
3. *Requirements Validation*: in a traditional requirements engineering process this activities are checking that the documented requirements and models are consistent and meet stakeholder needs. In HBMS an integrated requirements model should also be validated. But instead of a formal validation this could be done using simulation technology or converting the behavioral model into a human readable textual description.
4. *Requirements Management*: traditionally this means managing changes to the requirements as the system is developed and put into use. In HBMS people's behaviour can alter and therefore requirements change. So behavioral evolution has to be recognized, documented and managed.

In contrast to most traditional requirements engineering processes requirements in HBMS are not only building a model of a system to be developed but a model of individual knowledge to be used to customize an existing assistive system. Therefore a formal requirements description model and notation is essential. HBMS decided to elaborate a special Human Cognitive Model (HCM) based on KCPM (Klagenfurt Conceptual Predesign Model) as it was designed for reusing domain knowledge in requirements analysis [7,8].

To show the applicability of the proposed HBMS approach, a graphical modeling tool HCM Modeler was elaborated. It is based on Eclipse platform and therefore cross-platform and highly modular. There were elaborated extensions for model analysis including model transformation, model simulation and calculation of complexity and completeness metrics.

The HCM Modeler allows reusing existing model fragments, and provides the possibility to create more fine-grained models by creating sub-models. Behavioral models can be transformed to a textual list of requirements or to an individual ontology. The textual representation of the requirements are used to ensure that the model is correct and fits knowledge holder's needs. The individual ontology is used for integration and reasoning purposes supporting knowledge user later on.

The development process of HCM Modeler is still in progress. To enhance it the following activities are planned for the next future: refine existing requirements analysis and validation techniques, elaborate additional metrics to measure model's quality, implement the proposed model versioning and integrate semi-automatic procedures based on natural language processing of the textual descriptions to automate models creation.

References

1. Steg, H., Strese, H., Loroff, C., Hull, J., Schmidt, S.: Europe is facing a demographic challenge ambient assisted living offers solutions, pp. 1–85. European Societies (March 2006)
2. Cutchin, M.P.: The process of mediated aging-in-place: a theoretically and empirically based model. Social Science & Medicine 57(6), 1077–1090 (2003)
3. Brinkmann, M., Floeck, M., Litz, L.: Concept and design of an aal home monitoring system based on a personal computerized assistive unit. In: Mühlhäuser, M., Ferscha, A., Aitenbichler, E. (eds.) AmI 2007 Workshops. CCIS, vol. 11, pp. 218–227. Springer, Heidelberg (2008)
4. Grießer, A., Michael, J., Mayr, H.C.: Verhaltensmodellierung und automatisierte untersttzung im aal-projekt hbms. In: Technik für ein selbstbestimmtes Leben (AAL 2012), Berlin, 5p. (2012)
5. HBMS - Human Behaviour Monitoring and Support, `http://hbms-ainf.uni-klu.ac.at/` (retrieved on March 17, 2012)
6. Sommerville, I.: Software Engineering, 9th edn. Addison-Wesley, Harlow (2010)
7. Kop, C., Mayr, H.C.: Reusing domain knowledge in requirement analysis. In: Györkös, J. (ed.) 3rd Conference on Re-Engineering of Information Systems, RE 1994, Bled, pp. 133–147 (1994)
8. Kop, C., Mayr, H.: Conceptual predesign bridging the gap between requirements and conceptual design. In: Proceedings of the Third International Conference on Requirements Engineering, pp. 90–98 (1998)

Author Index