José Cordeiro
Leszek A. Maciaszek
Joaquim Filipe (Eds.)

LNBIP 141

Enterprise Information Systems

14th International Conference, ICEIS 2012
Wroclaw, Poland, June/July 2012
Revised Selected Papers

Springer

Lecture Notes in Business Information Processing 141

Series Editors

For further volumes:
http://www.springer.com/series/7911

José Cordeiro · Leszek A. Maciaszek
Joaquim Filipe (Eds.)

Enterprise Information Systems

14th International Conference, ICEIS 2012
Wroclaw, Poland, June 28–July 1, 2012
Revised Selected Papers

Springer

Editors
José Cordeiro
Joaquim Filipe
Institute for Systems and Technologies of Information, Control and Communication
Setúbal
Portugal

and

Instituto Politécnico de Setúbal
Setúbal
Portugal

Leszek A. Maciaszek
Wroclaw University of Economics
Wroclaw
Poland

and

Macquarie University
Sydney
Australia

ISSN 1865-1348 ISSN 1865-1356 (electronic)
ISBN 978-3-642-40653-9 ISBN 978-3-642-40654-6 (eBook)
DOI 10.1007/978-3-642-40654-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013950735

Printed on acid-free paper

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

Preface

The present book includes extended and revised versions of a set of selected papers from the 14th International Conference on Enterprise Information Systems (ICEIS 2013), held in Wroclaw, Poland, during June 28–July 1, 2012 and co-organized by the Wroclaw University of Economics. The conference was sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC), held in cooperation with the Association for Advancement of Artificial Intelligence (AAAI), the Institute of Electronics Information and Communication Engineers (IEICE), the IEICE Special Interest Group on Software Interprise Modelling (SWIM), the Association for Computation Machinery (ACM), the ACM Special Interest Group on Management Information Systems (SIGMIS), and the ACM Special Interest Group on Computer Human Interaction.

The conference was organized in six simultaneous tracks: "Databases and Information Systems Integration, Artificial Intelligence and Decision Support Systems, Information Systems Analysis and Specification, Software Agents and Internet Computing, Human–Computer Interaction, and Enterprise Architecture". The book is based on the same structure.

ICEIS 2012 received 299 paper submissions, from 49 countries and districts on all continents. From these, after a blind review process, only 28 were accepted as full papers, of which 25 were selected for inclusion in this book, based on the classifications provided by the Program Committee. The selected papers reflect state-of-the-art research work that is often oriented towards real world applications and highlights the benefits of Information Systems and Technology for industry and services, thus making a bridge between the academia and the enterprise worlds. These high quality standards will be maintained and reinforced at ICEIS 2013, to be held in Angers, France, and in future conferences.

Furthermore, ICEIS 2012 included 5 plenary keynote lectures given by Schahram Dustdar (Vienna University of Technology, Austria), Dimitris Karagiannis (University of Vienna, Austria), Steffen Staab (University of Koblenz-Landau, Germany), Pericles Loucopoulos (Loughborough University, UK), and Yannis Manolopoulos (Aristotle University, Greece). We would like to express our appreciation to all of them and in particular, to those who took the time to contribute a paper to this book.

On behalf of the conference Organizing Committee, we would like to thank all participants. First of all the authors, whose quality work is the essence of the conference and the members of the Program Committee, who helped us with their expertise and diligence in reviewing the papers. Also we all know, producing a conference requires the effort of many individuals. We also wish to thank all the members of our Organizing Committee, whose work and commitment were invaluable.

March 2012

José Cordeiro
Leszek A. Maciaszek
Joaquim Filipe

Organization

Conference Chair

Joaquim Filipe	Polytechnic Institute of Setúbal/INSTICC Portugal

Program Co-chairs

José Cordeiro	Polytechnic Institute of Setúbal/INSTICC Portugal
Leszek A. Maciaszek	Wroclaw University of Economics Poland/Macquarie University ~ Sydney Australia
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria Italy

Organizing Committee

Helder Coelhas	INSTICC Portugal
Andreia Costa	INSTICC Portugal
Patrícia Duarte	INSTICC Portugal
Bruno Encarnação	INSTICC Portugal
Liliana Medina	INSTICC Portugal
Raquel Pedrosa	INSTICC Portugal
Vitor Pedrosa	INSTICC Portugal
Daniel Pereira	INSTICC Portugal
Cláudia Pinto	INSTICC Portugal
Susana Ribeiro	INSTICC Portugal
José Varela	INSTICC Portugal
Pedro Varela	INSTICC Portugal

Senior Program Committee

Balbir Barn	UK
Albert Cheng	USA
Jan Dietz	The Netherlands
Schahram Dustdar	Austria
António Figueiredo	Portugal
Michel Léonard	Switzerland
Kecheng Liu	UK

Pericles Loucopoulos	UK
Andrea de Lucia	Italy
Yannis Manolopoulos	Greece
Masao Johannes Matsumoto	Japan
Alain Pirotte	Belgium
Klaus Pohl	Germany
Matthias Rauterberg	The Netherlands
Colette Rolland	France
Narcyz Roztocki	USA
Abdel-Badeeh Mohamed Salem	Egypt
Bernadette Sharp	UK
Alexander Smirnov	Russian Federation
Ronald Stamper	UK
Merrill Warkentin	USA

Program Committee

Miguel Angel Martinez Aguilar	Spain
Patrick Albers	France
Rainer Alt	Germany
Vasco Amaral	Portugal
Andreas S. Andreou	Cyprus
Wudhichai Assawinchaichote	Thailand
Cecilia Baranauskas	Brazil
Senén Barro	Spain
Rémi Bastide	France
Bernhard Bauer	Germany
Lamia Hadrich Belguith	Tunisia
Jorge Bernardino	Portugal
Felix Biscarri	Spain
Oliver Bittel	Germany
Danielle Boulanger	France
Jean-Louis Boulanger	France
Peter Busch	Australia
Miguel Calejo	Portugal
Coral Calero	Spain
Luis M. Camarinha-Matos	Portugal
Olivier Camp	France
Roy Campbell	USA
Gerardo Canfora	Italy
Manuel Isidoro Capel-Tuñón	Spain
Angélica Caro	Chile
Jose Jesus Castro-schez	Spain
Luca Cernuzzi	Paraguay

Sergio de Cesare	UK
Ming-Puu Chen	Taiwan
Shiping Chen	Australia
Shu-Ching Chen	USA
Max Chevalier	France
Witold Chmielarz	Poland
Daniela Barreiro Claro	Brazil
Cesar Collazos	Colombia
Jose Eduardo Corcoles	Spain
Karl Cox	UK
Sharon Cox	UK
Alfredo Cuzzocrea	Italy
Maria Damiani	Italy
Vincenzo Deufemia	Italy
Kamil Dimililer	Cyprus
José Javier Dolado	Spain
Dulce Domingos	Portugal
César Domínguez	Spain
Ming Dong	China
Juan C. Dueñas	Spain
Hans-Dieter Ehrich	Germany
João Faria	Portugal
Antonio Fariña	Spain
Jamel Feki	Tunisia
Antonio Fernández-Caballero	Spain
Edilson Ferneda	Brazil
Maria João Silva Costa Ferreira	Portugal
Paulo Ferreira	Portugal
Filomena Ferrucci	Italy
Rita Francese	Italy
Bogdan Franczyk	Germany
Ana Fred	Portugal
Lixin Fu	USA
Mariagrazia Fugini	Italy
Jose A. Gallud	Spain
Matjaz Gams	Slovenia
Maria Ganzha	Poland
Juan Garbajosa	Spain
Mouzhi Ge	Germany
Marcela Genero	Spain
Joseph Giampapa	USA
Raúl Giráldez	Spain
Pascual Gonzalez	Spain
Robert Goodwin	Australia
Feliz Gouveia	Portugal

Luis Borges Gouveia	Portugal
Virginie Govaere	France
Janis Grabis	Latvia
Maria Carmen Penadés Gramaje	Spain
Sven Groppe	Germany
Wieslawa Gryncewicz	Poland
Nuno Guimarães	Portugal
Maki K. Habib	Egypt
Yaakov Hacohen-Kerner	Israel
Sylvain Hallé	Canada
Slimane Hammoudi	France
Wahab Hamou-Lhadj	Canada
Christian Heinlein	Germany
Markus Helfert	Ireland
Suvineetha Herath	USA
Orland Hoeber	Canada
Wladyslaw Homenda	Poland
Jun Hong	UK
Wei-Chiang Hong	Taiwan
Miguel J. Hornos	Spain
Kai-I Huang	Taiwan
Akram Idani	France
Marta Indulska	Australia
François Jacquenet	France
Arturo Jaime	Spain
Marijn Janssen	The Netherlands
Wassim Jaziri	Tunisia
Paul Johannesson	Sweden
Jan Jürjens	Germany
Michail Kalogiannakis	Greece
Nikos Karacapilidis	Greece
Nikitas Karanikolas	Greece
Stamatis Karnouskos	Germany
Andrea Kienle	Germany
Marite Kirikova	Latvia
Alexander Knapp	Germany
Fotis Kokkoras	Greece
Ryszard Kowalczyk	Australia
John Krogstie	Norway
Subodha Kumar	USA
Rob Kusters	The Netherlands
Halina Kwasnicka	Poland
Alain Leger	France
Kauko Leiviskä	Finland
Daniel Lemire	Canada

Da-Yin Liao	USA
Luis Jiménez Linares	Spain
Panos Linos	USA
Stephane Loiseau	France
João Correia Lopes	Portugal
Maria Filomena Cerqueira de Castro Lopes	Portugal
María Dolores Lozano	Spain
Miguel R. Luaces	Spain
André Ludwig	Germany
Vicente Luque-Centeno	Spain
Mark Lycett	UK
Lukasz Lysik	Poland
Cristiano Maciel	Brazil
Rita Suzana Pitangueira Maciel	Brazil
Mirko Malekovic	Croatia
Nuno Mamede	Portugal
Paolo Maresca	Italy
Pierre Maret	France
Tiziana Margaria	Germany
Farhi Marir	UK
Herve Martin	France
Maria João Martins	Portugal
Katsuhisa Maruyama	Japan
Viviana Mascardi	Italy
David Martins de Matos	Portugal
Wolfgang Mayer	Australia
Javier Medina	Spain
Jerzy Michnik	Poland
Luo Ming	Singapore
Michele Missikoff	Italy
Ghodrat Moghadampour	Finland
Pascal Molli	France
Lars Mönch	Germany
Valérie Monfort	Tunisia
Francisco Montero	Spain
Carlos León de Mora	Spain
Paula Morais	Portugal
Fernando Moreira	Portugal
Haralambos Mouratidis	UK
Pietro Murano	UK
Ana Neves	Portugal
Matthias Nickles	Germany
Ann Nosseir	Egypt
Jose Angel Olivas	Spain
David Olson	USA

Mehmet Orgun	Australia
Andrés Muñoz Ortega	Spain
Samia Oussena	UK
Sietse Overbeek	The Netherlands
Tansel Ozyer	Turkey
Claus Pahl	Ireland
Marcin Paprzycki	Poland
José R. Paramá	Spain
Eric Pardede	Australia
Viviana Patti	Italy
Loris Penserini	Italy
Massimiliano Di Penta	Italy
Laurent Péridy	France
Dana Petcu	Romania
Yannis A. Phillis	Greece
Josef Pieprzyk	Australia
Selwyn Piramuthu	USA
José Pires	Portugal
Luís Ferreira Pires	The Netherlands
Geert Poels	Belgium
Michal Polasik	Poland
Srini Ramaswamy	India
T. Ramayah	Malaysia
Pedro Ramos	Portugal
Marek Reformat	Canada
Francisco Regateiro	Portugal
Hajo A. Reijers	The Netherlands
Ulrich Reimer	Switzerland
Marinette Revenu	France
Nuno de Magalhães Ribeiro	Portugal
Debbie Richards	Australia
Michele Risi	Italy
David Rivreau	France
Alfonso Rodriguez	Chile
Daniel Rodriguez	Spain
Pilar Rodriguez	Spain
Oscar Mario Rodriguez-Elias	Mexico
Erik Rolland	USA
Jose Raul Romero	Spain
David G. Rosado	Spain
Gustavo Rossi	Argentina
Artur Rot	Poland
Francisco Ruiz	Spain
Roberto Ruiz	Spain
Belen Vela Sanchez	Spain

Luis Enrique Sánchez	Spain
Manuel Filipe Santos	Portugal
Jurek Sasiadek	Canada
Andrea Schaerf	Italy
Daniel Schang	France
Sissel Guttormsen Schär	Switzerland
Isabel Seruca	Portugal
Jianhua Shao	UK
Hossein Sharif	UK
Alberto Silva	Portugal
Sean Siqueira	Brazil
Spiros Sirmakessis	Greece
Hala Skaf-molli	France
Chantal Soule-Dupuy	France
José Neuman de Souza	Brazil
Martin Stanton	UK
Chris Stary	Austria
Dick Stenmark	Sweden
Stefan Strecker	Germany
Vijayan Sugumaran	USA
Hiroki Suguri	Japan
Lily Sun	UK
Raj Sunderraman	USA
Jerzy Surma	Poland
Gion K. Svedberg	Sweden
Ryszard Tadeusiewicz	Poland
Vladimir Tarasov	Sweden
Arthur Tatnall	Australia
Sotirios Terzis	UK
Claudine Toffolon	France
Theodoros Tzouramanis	Greece
Athina Vakali	Greece
José Ângelo Braga de Vasconcelos	Portugal
Michael Vassilakopoulos	Greece
Christine Verdier	France
François Vernadat	France
Maria Esther Vidal	Venezuela
Aurora Vizcaino	Spain
Bing Wang	UK
Dariusz Wawrzyniak	Poland
Hans Weghorn	Germany
Hans Weigand	The Netherlands
Gerhard Weiss	The Netherlands
Graham Winstanley	UK
Wita Wojtkowski	USA

Viacheslav Wolfengagen	Russian Federation
Andreas Wombacher	The Netherlands
Robert Wrembel	Poland
Stanislaw Wrycza	Poland
Min Wu	USA
Wen-Yen Wu	Taiwan
Mudasser Wyne	USA
Haiping Xu	USA
Sadok Ben Yahia	Tunisia
Lili Yang	UK
Ping Yu	Australia
Yugang Yu	The Netherlands
Wei Zhou	France
Eugenio Zimeo	Italy
Lin Zongkai	China

Auxiliary Reviewers

Ankica Barisic	Portugal
Bruno Barroca	Portugal
Gabriele Bavota	Italy
Paulo Carreira	Portugal
Jan Claes	Belgium
Rui Domingues	Portugal
Jessica Diaz Fernandez	Spain
Daniel Lopez Fernnadez	Spain
Rodrigo Garcia-Carmona	Spain
Laura Sánchez González	Spain
Krzysztof Kania	Poland
Sandra Lovrencic	Croatia
Paloma Cáceres García de Marina	Spain
Luã Marcelo Muriana	Brazil
David Musat	Italy
Álvaro Navas	Spain
Annibale Panichella	Italy
Ignazio Passero	Italy
Jonas Poelmans	Belgium
Hércules Antonio do Prado	Brazil
Abdallah Qusef	Italy
Federica Sarro	Italy
Markus Schatten	Croatia
Claudio Schifanella	Italy
Diego Seco	Spain

Invited Speakers

Schahram Dustdar	Vienna University of Technology, Austria
Dimitris Karagiannis	University of Vienna, Austria
Steffen Staab	University of Koblenz-Landau, Germany
Pericles Loucopoulos	Loughborough University, UK
Yannis Manolopoulos	Aristotle University, Greece

Contents

Artificial Intelligence and Decision Support Systems

Information Systems Analysis and Specification

Software Agents and Internet Computing

Part I

Invited Papers

An Engineering Approach for the Design of Hybrid Modelling Methods

Dimitris Karagiannis(✉) and Margit Schwab

Universitaet Wien, Research Group Knowledge Engineering,
Waehringer Strasse 29, 1090 Vienna, Austria
{dk, ms}@dke.univie.ac.at

Abstract. A fast moving business environment requires flexible and open conceptual modelling approaches for the discussion of diverse needs from the business point of view. In the conception and design of these business needs and resulting requirements, manifold modelling languages and methods on different levels of the realisation process are available. Despite their number, flexibility in this paper is referred to the exchange of 'building blocks' of modelling frameworks and the composition of *hybrid modelling methods*. That this claim for flexibility does neither affect efficiency nor goes at the account of a sound conception of the hybrid modelling method, a deliberate procedure comprising different steps is required. We call this procedure *conceptualisation*. At the end of this procedure an implementation on a meta-modelling platform is performed and the result of the conceptualisation process is a 'deployable tool'. Furthermore we imply the hybrid modelling method is of a graphical, semi-formal kind and based on a meta-modelling approach. The conceptualisation is a platform dependent activity. In the paper at hand the meta-modelling platform ADOxx® is used.

Keywords: Meta-model · Modelling language · Hybrid modelling · Adoxx® · Meta-modelling platform · Method engineering

1 Introduction

In the disciplines of informatics and business informatics models form in many cases the methodological basis, but represent in particular the connection to reality. Depending on the scenario underlying the model construction some of the situations that are described in form of models are known. In a diversity of 'developing situations' the design of models is required because reality itself does not yet exist [9]. In both cases a deliberate consideration of the described concepts as well as expected behaviour is insofar of essential relevance. Aligned to these considerations and resulting requirements the modelling language intended to be used for the model construction must offer an appropriate set of expressions and incorporate topic related concinnity. The more expressive the modelling language is the more detailed information can be modelled. The better the abstraction of generic modelling concepts succeeds the more expressive is the language.

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 3–17, 2013.
DOI: 10.1007/978-3-642-40654-6_1,

Based on the initial requirements the different modelling methods are developed for a specific domain and for a particular purpose. In a ‘reuse scenario’ of the modelling language where the application scenario respectively the purpose of the models vary, not all of the modelling concepts the language offers, may be meaningful for or meet the needs of ‘the new scenario’. This is even true for basically generically held modelling languages. A surplus of modelling concepts is certainly a negligible ‘annoyance’, if there are too little or semantically different modelling concepts as required, then this is graver. The development of a new modelling language could solve this issue. Another scenario is that the missing concepts can be regained in another modelling language respectively modelling method. In such a case a new development might not seem feasible, e.g. too laborious, but the composition of a ‘user-need-specific’ modelling method is a tempting approach. This is due to several reasons for example the availability of established modelling concepts with an already clear recognition and clear in their semantic interpretation, an explicit usage scenario with an already determined high level aggregated versus micro-flow knowledge representation, an already solved solution approach for a certain problem, and once established – the alignment to the existing knowledge base, eventually existing implementation examples in form of tools, etc.

In the paper we speak of *hybrid modelling* for describing the process of creation of such a ‘user-need-specific’ modelling method. We assume an actual modelling need and therefore discuss this process from the angle of ‘implementation entailing preparation steps’, i.e. the *conceptualisation*. We will highlight the fundamental integration problems of this undertaking and discuss them from a meta-model point of view, from syntactical, structural and semantic heterogeneity. We use the meta-modelling framework of Karagiannis/Kühn for the definition and distinction of the different parts of a modelling method [5].

The paper is organized as follows. Section 2 is devoted to explications on hybrid modelling and resulting integration challenges. Section 3 describes the actual procedure for hybrid modelling in form of a conceptualisation life cycle. Section 4 concludes the paper and gives an outlook on further research questions to be addressed.

2 Hybrid Modelling

The topic of hybrid modelling is predominantly related to two major disciplines – conceptual and enterprise modelling. In the informatics domain conceptual modelling has first emerged for the description and design of databases in form of semantic data models, of programming languages in form of object-orientation and for example UML and of artificial intelligence, AI in form of knowledge representation, e.g. description logics [2, p. 4]. The development of conceptual modelling shows that it was at its beginning a sole computer science topic but got increasingly adopted in enterprise modelling at this stage. In enterprise modelling scenarios the angle on information that is represented in form of models is more related to business and management needs and adds in particular the ‘process perspective’ [11]. In both strands distinct modelling languages offering appropriate modelling concepts have been developed for expressing the specific requirements of the different domain.

Traditionally conceptual modelling languages are rather data-structure-oriented, a property that represents the major difference in contrast to enterprise modelling. In enterprise modelling the need is to represent business processes and the way how to process content from a business logic view in form of models. This need requires other modelling concepts [4, pp. 12–20].

An increasingly powerful IT-support enables gradually more comprehensive and complex modelling approaches. In order to facilitate the business needs with flexible and deliberate IT-solutions modelling frameworks tend to offer a vertical integration of different 'conceptional' levels, i.e. from a high level strategic goal definition to the actual micro flow representation within a specific IT system. Hand in hand with these comprehensive approaches goes that the actual 'end user' of these modelling frameworks must show comprehensive knowledge about the 'intended use' of these frameworks also. Still, in most cases the offered modelling frameworks provide a certain direction with the supported application scenarios, e.g. process versus actor-goal oriented modelling approaches. We claim that due to diverse business requirements in a fast moving environment amplified flexibility and open modelling approaches in form of a 'modular construction system' are of need – the hybrid modelling approach.

Despite, the meaning of hybrid defined in the Oxford English Dictionary as "[…] anything derived from heterogeneous sources, or composed of different or incongruous elements; in *Philol.* a compound formed of elements belonging to different languages […]" the resulting hybrid modelling method must fulfil certain formalisms [10].

The first requirement is that the result of hybrid modelling is a modelling method that falls in the category of a graphical semi-formal modelling method based on a meta-modelling approach. This output-related requirement already claims certain formalisms regarding the composition and structure of the modelling method.

A second requirement is that the hybrid modelling method is implemented and offered in form of a modelling tool. The implementation itself uses the programming language/s as formalism/s. The better the resulting hybrid modelling method is conceptually composed, the smoother the transformation of the concepts to the codes can be done. The requirement of an implementation is based on the conviction that an efficient modelling support can only be guaranteed by a modelling tool. Hand in hand with this persuasion goes the assertion that the user requires a structured procedure for creating models that are 'machine-processable', e.g. for corresponding evaluation algorithms. In the latter case we do not focus on the creation of executable code from the conceptual model.

Adopting modelling concepts from different modelling frameworks entails several challenges on different levels regarding their integration. For all further explanations the focus is on the metamodel of language levels as depicted in Fig. 1. The classification is according to Karagiannis/Kühn [5, p. 453].

Although, the aimed structure shows a metamodel formalism, the different 'building blocks' can be of different formalisms, e.g. logic-based formal grammar. Furthermore they can exhibit diverse abstraction levels. On a metamodel level this brings along the following states in a way that metamodels vary

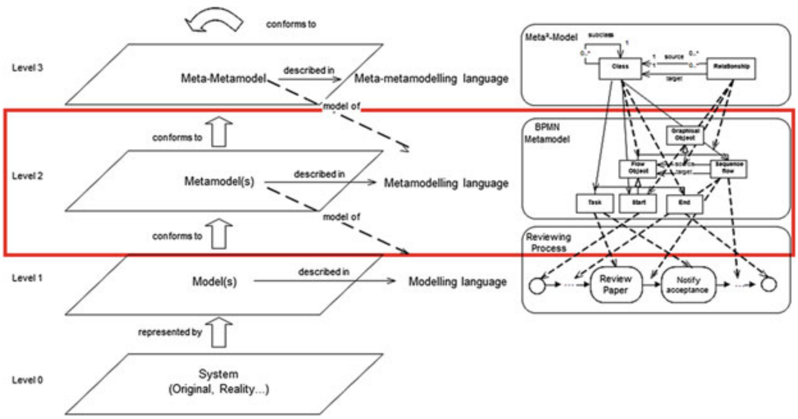


Fig. 1. Focus on the Meta-Model of Language Levels [5].

- Vertically, showing different levels of detail;
- Horizontally, the modelling concepts are on the same abstraction level but describe different aspects;
- In both ways.

One of the most essential parts in hybrid modelling is the integration of the modelling concepts and the translation from dissimilar formalism on the metamodel level. As on the metamodel level not at least as modelling language inherent constraints are determined in this integration, e.g. which association connects which modelling elements. The intricacies of a modelling method are rather found on the abstract syntax and semantic parts that also come along with the expressiveness and dynamics of the notation. This implies that the integration on a metamodel level is insufficient.

Though, the structure and constraints of a modelling language are often in focus, a modelling method provides further concepts apart from its modelling language. At first extent the 'functionality' a modelling method offers is not obvious but it becomes effective once implemented 'as a piece of software'. In this context we would like to stress mechanisms and algorithms. These are characterised by the fact that they express a form of dynamic behaviour of the modelling method and become applicable in the composition of the actual 'end user models' as instances created by means of the modelling method. Assuming that an algorithm is part of the hybrid modelling method, it requires an appropriate 'translation' of its general behaviour and semantic behaviour to the new modelling method, e.g. attributes containing input values.

For all the adopted parts and pieces apply that they were initially designed from a particular angle, the one of the initial method developer, in order to convey particular semantics and demand a particular handling in their effective use. Assuming the hybrid modelling approach does not mean a reuse of existing concepts for a further development, the initial meaning needs to be taken into account in essential accuracy.

3 The Conceptualisation Life Cycle for Hybrid Modelling

In the process of hybrid modelling, we identify three basic phases. These are the creation phase, the design phase and the compilation phase and form together the conceptualisation life cycle that is shown in Fig. 2. The first phase is related to the application scenario and the need of the user and refers basically to the selection process the user performs for identifying the existing modelling concepts within the hybrid method. The second phase is related to preparatory steps for the implementation. The most essential task is the determination of the meta-modelling platform.

The design phase cannot be done without knowing the target platform. We will concentrate on describing the tasks of that phase in the sequel. The compilation phase relates to the creation of the modelling tool. Depending on the deployment strategy different solutions are conceivable, e.g. standalone with web-access, mobile app, etc.

3.1 The Creation Phase

At the starting point of this phase the actual application scenario for which models should be created, must be clear. Based on this need the different building blocks for the hybrid modelling method can be determined. The detailed study and analysis of the selected building blocks goes hand in hand with this determination. The actual integration of the selected parts is understood as 'a merge' of existing concepts. The result of the creation phase is a detailed picture of the available hybrid modelling concepts and their dependencies, required mechanisms and algorithms including eventual requirements with regards to 'model processing' on an instance level, like user triggered data actualisations, report generation, dynamic visualisations. The detailed picture needs to be described in a way that the actual intended usage of the modelling concepts can be understandable. Depending on the maturity of the underlying building blocks the description can be of formal but also informal kind or can for example include the definition of a consistent meta-model of the hybrid modelling method on a sole conceptual level. The more concrete the description in the creation phase is, the easier is the engineering work within the subsequent design phase and the more aligned to the initial purpose of the hybrid modelling undertaking the resulting

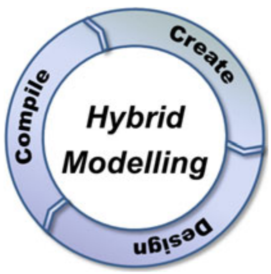


Fig. 2. Conceptualisation Life Cycle for Hybrid Modelling.

method will be. Independent of the level of formalism in the creation of a hybrid modelling method the three major parts must be addressed:

- Modelling language including syntax, semantics and notation,
- Mechanisms and algorithms and
- Modelling procedure comprising the actual usage of the modelling language as intended by the 'hybrid modelling method developer', i.e. method engineer.

Several challenges regarding transformation and integration emerge from bringing the different building blocks on a consistent level. From the angle of the method engineer this integration task is easier if the selected method and the selected platform provide the same concepts, e.g. a meta-modelling-approach.

Meta-models are not able to express all syntactical rules that have to be taken into account when creating a valid instance model that is conform to the defined modelling language [7, p. 68]. For this reason a detailed description of the three parts of a modelling language - the syntax, the semantic and notation - is of need. On this detailed level design decisions are required. These can be necessary due to the following points:

- *Syntactical heterogeneity,* which represent the difference in formats intended for the serialization of metamodels.
- *Structural heterogeneity:*
  - *Representational heterogeneity:* metamodels are represented using different metamodelling languages, each of them showing difference in its expressive power of available modelling primitives (classes, attributes, …);
  - *Schematic heterogeneity:* equal concepts are modelled either with different modelling primitives or with different number of primitives.
- *Semantic heterogeneity,* which represents differences in the meaning of the considered metamodel concepts.

Last but not least the 'look and feel' in the notations requires integration. Though it may seem the lesser part, for the untrained addressee with regards to the handling and reading of diagrammatic models, colour is for example an important distinction element to keep the available shapes and as a consequence their meaning apart. A consistent notation is a design decision from the usability point of view.

Depending on the design decisions made in the modelling language part of the hybrid modelling method, the parts 'mechanisms and algorithms' and the 'modelling procedure' part must be aligned.

3.2 The Design Phase

In the design phase the preconditions for the later implementation respectively customisation phase are elaborated. Depending on the 'degree of maturity', i.e. how many and well elaborated modelling concepts it offers, underlying formal definition, etc. of a given modelling method the design phase is more or less extensive.

The prerequisites for starting this phase are on the one hand the input from the creation phase and on the other hand the platform the hybrid modelling method is realised on. The hybrid modelling method at this stage is in a status in the range from 'the raw script' to 'ready-to-use' in its theoretical description and explains the method from a "business logic" point of view depending on the underlying domain. The meta-modelling platform, in this case ADOxx® adds the 'technical' point of view. The entire conceptualisation process cannot be done if the platform is not determined. The design phase of the conceptualisation life cycle shows two elementary results: a conceptualisation meta-model and an implementation meta-model.

3.2.1 The Meta-Models

The metamodel term in this paper is understood as the description model of a language, i.e. the modelling language. The metamodel comprises in itself a particular structure in form of rules and constraints how the modelling concepts are intended to be combined by the initial method developer. The metamodel is thus the grammar of the modelling language [12].

In contrast to the pure language metamodel there are the – in general generically held – concepts of the meta-modelling platform, so the meta^{2}model, i.e. level 3 of the graphic in Fig. 1. The particular challenge now is to map the language metamodel to the 'generic' platform concepts. We state that this is a critical task and the better this alignment succeeds the better the applicability and the more flexible the handling of the object language will be at the later stage.

Assuming a top-down approach a comprehensive analysis of the to-be-implemented modelling concepts is the first step and includes the identification of hybrid modelling method inherent dependencies and constraints. If a pure language metamodel is lacking, it is likely that the actual 'immediate' mapping of the modelling concepts including their dependencies respectively constraints gives a different result within the conceptualisation life cycle. The reason for this is, that 'hierarchy of modelling concepts' with regards to the platform metamodel can be different than from a sole 'business logic' point of view.

In any case the representation of the language metamodel in the platform metamodel requires design decisions. These are not necessarily due to an inaccurate description or specification of the hybrid language metamodel but related to the fact that some questions arise the first time in the preparation of its implementation. These questions are triggered by a combination of the structure of the platform and the later usability of the method. A typical example of such 'gaps' are the description of associations and the constraints they impose. As a popular description language for metamodels works the UML class diagram and although the cardinalities for the classes representing modelling concepts are frequently indicated, time and again the dependencies introduced due to for example inheritance assumptions are paid little attention. For the implementation the level where association-related constraints become effective are important. If the language metamodel is not expressive enough and if the method developer does not give any further information, only the analysis of eventually existing example models show the actual use of the respective modelling concepts including 'theoretically' integrated modelling constraints. Therefore inheritance and hierarchical concepts of both the language and the platform metamodel represent a source of additional design decisions due to the aimed implementation. There are further sources that are addressed in the sequel by means of the metamodel in a specific platform, i.e. the ADOxx® platform meta^{2}model [3, 7].[1]

Although, the analysis of the language and the platform metamodels gives information about the hybrid modelling method inherent dependencies and constraints, the actual parts of the hybrid modelling language require further attention. The conceptualisation of the notation and semantic part and from them an alignment of the syntax part, grants to the modelling language increased expressiveness and flexible handling with regards to conveying intricacies of information. Therefore the conceptualisation of the language meta-model is only part of the rent.

3.2.2 Notation, Syntax and Semantic

The structure of a graphical, semi-formal hybrid modelling method consists of three parts, the notation, the syntax and the semantic.

The Notation: The notation describes the graphical representation of the elements of the modelling language. Although the notation is very often seen as the least important part, it is a very obvious aspect as it is visualised. So, if for example the notation is not specified detailed enough, different interpretations depending on the method engineer's perception and design of the modelling concept are possible. This has for example happened in case of the *i** modelling method. Fig. 3 shows a selection of two modelling concepts originating from the *i** modelling method. Interestingly the shape of the 'Actor' modelling concept is rather consistent whereas the shape of the 'Softgoal' modelling concept is difficult to reach [13, 15].

For the definition of the notation for a hybrid modelling method a similar situation is conceivable. The following combinations are likely, either there are

- More than one modelling classes for the same modelling concept providing different notations;
- Modelling concepts where the notation part has not yet been defined.

In both cases a design decision is required how to proceed for the hybrid modelling method. This design decision is again ideally done by the 'hybrid modelling language developer' in the creation phase already.

Summarising the said, the behaviour of the different classes with regards to their graphical representation is determined when specifying the notation of the hybrid modelling language. The notation is moreover related with the semantics and the syntax in a way that the graphical representation is determined by a specific attribute

[1] ADOxx® is the metamodelling platform that is used by the Research Group Knowledge Engineering of the University of Vienna for their research on metamodels, metamodelling and model languages. It is an extensible, repository-based platform, offers a three-step modelling hierarchy and is based on a rich meta-model. ADOxx® is a 'development' platform for modelling languages which are founded on a metamodel approach. Karagiannis, D.; Visic, N. (2011): "Next Generation of Modelling Platforms". In: Grabis, J.; Kirikova, M. (eds.): Perspectives in Business Informatics Research, Vol. 90, pages 19–28, Springer Berlin Heidelberg. Furthermore the projects of the Open Models Initiative are realised on this platform (http://www.openmodels.at).

Modelling Class : Actor				
Notation				
Object sizing	not sizeable	sizeable	sizeable	not sizeable
Adaptive notation	no	no	colour setting on object level	'name' visualisation setting on object level
Modelling Language	OpenOME	TAOM4E	jUCMNav/GRL	*i** on ADOxx

Modelling Class : Softgoal				
Notation				
Object sizing	sizeable	sizeable	not sizeable	not sizeable
Adaptive notation	no	no	colour triggered by attribute value	icon visualisation triggered by attr. value
Modelling Language	OpenOME	TAOM4E	jUCMNav/GRL	*i** on ADOxx

Fig. 3. Examples of Different Notations for the Same Modelling Concept.

value. The attribute value is part of the semantic whereas the attribute itself is part of the syntax.

The Semantic: The semantics describe the meaning of the modelling concepts of a modelling language, i.e. contrasting the objects in reality and which are to be mapped in the model and how the language elements have to be interpreted. The semantics are also expressed in the values the attributes defined in the syntax part can adopt. The graphical representation of the respective modelling concept eventually underlines the semantics. In the conceptualisation of a hybrid modelling method a clear delineation of the semantics of modelling concepts is of need. For a better explanation the discussion should be done by means of an example.

The hybrid modelling method should contain modelling concepts of the *i** method and of a business process modelling method, BPMS. Although the integration of parts of these two modelling languages is rather conceivable on a vertical level due to the general purpose of the modelling methods, some modelling concepts require special attention. In the *i** modelling method the 'view' of a strategic dependency model contains the modelling concepts of Actor, Agent, Role, Position, Dependency Link, Association Link and the 'intentional elements'. In the BPMS method there is the model type working environment model containing the modelling concepts of Organisational Unit, Performer, Role, Position, Is subordinated, Belongs to, Has role, Is manager and Has position. Selected elements of both modelling methods are given in Fig. 4.

Even if one would integrate the strategic dependency model and from the BPMS method the model type business process model, so sets of modelling concepts with different predetermined purposes, in the business process model consideration the Role and or Performer is a central element for the specification on which level respectively skill level the activities within the business process should be processed.

Modelling Concepts		*i* classes and relations*					
		Actor	**Agent**	**Role**	**Position**	**Association Link**	**Dependency Link**
BPMS classes and relations	**Organisational unit**	~	!=	~	!=	-	-
	Performer	~	~ almost 1:1	!=	!=	-	-
	Role	~	!=	~ almost 1:1	~	-	-
	Position	!=	!=	!=	!=	-	-
	Is subordinated	-	-	-	-	~	!=
	Belongs to	-	-	-	-	~	!=
	Has role	-	-	-	-	~	!=
	Is manager	-	-	-	-	~	!=
	Has postion	-	-	-	-	~	!=

Caption:	
!=	unlike, does not correspond at all
1:1	identical in their natural language description and use
~	natural language description and use show similarities
-	not applicable - comparision of modelling class to reation class

Fig. 4. Contrasting Modelling Concepts from the *i** and the BPMS Method.

In order to determine if for example the Role in the *i** modelling method can be reused for the business process models the assigned semantic from the description and usage interpretation of the modelling concepts is of need.

What can be learnt from Fig. 4 is that those elements that show similarities require a clear interpretation for avoiding a 'muddling through' of the modelling concepts in their actual use on an object level.

The Syntax: The syntax describes the dependencies and constraints in between the modelling concepts and is furthermore represented in the description of the properties of these in form of attributes. Almost every modelling concept offers a 'name' attribute. Other attributes are used for a comprehensive description of the domain and application scenario the modelling method is used for. An integration of attributes of semantically identical modelling concepts of two different modelling methods will rather be the creation of the 'common multiple'. In the modelling concept of a Role in the *i** method only the attribute 'name' was defined; in the Role of the BPMS method besides the 'name' attribute seventeen further attributes are defined for a comprehensive description of the modelling concept. Moreover three of them trigger by means of predefined values a change of the graphical notation [1].

The detailed steps of the design phase of the conceptualisation life cycle for hybrid modelling are discussed by Xu et al. using modelling concepts of *i** and UML to merge them to *Active i**. The *Active i** modelling method is a representative example that has emerged from hybrid modelling [14].

3.2.3 Mechanisms and Algorithms

The conceptualisation of the hybrid modelling language was only described for the language part so far. The underlying modelling framework of Karagiannis/Kühn requires the modelling procedure and algorithms and mechanisms as integrated modelling method parts. The modelling procedure becomes manifest in the composition of the hybrid modelling language, e.g. the definition of model types and an accompanying instruction manual for example in form of language-specific modelling guidelines. The mechanisms and algorithms are of different kind as they usually represent extensions of the modelling language respectively of the usage scenario of the modelling language. In this understanding the mechanisms and algorithms necessitate consideration regarding parameters which form their input and a consideration in which way they are realised in the modelling language, e.g. which attribute of what attribute type contains values required for a composition algorithm. Furthermore we learn from the modelling framework of Karagiannis/Kühn that three different mechanisms and algorithms are distinguished by the criterion if they are specific for the modelling technique or if they are generic, e.g. breadth-first-search algorithm. If the algorithms work with the modelling concepts of the hybrid modelling language, it has to be defined if the algorithms require an alignment of the syntax and related semantics parts, e.g. in form of additional attribute values to gather input values or to capture calculation results from algorithms. Further examples where algorithms affect directly the modelling concepts are if constraints are triggered by means of algorithms, e.g. in form of the verification of modelling scenarios, cardinality checks. The design of such functionality-related algorithms is essential for the 'smooth and easy going' use of the implemented hybrid modelling method and is of high significance within the conceptualisation life cycle.

The fact that the platform itself offers general predefined functionality, e.g. report generation or 'only' interfaces where additional functionality can be integrated in the general language composition forms another aspect in the conceptualisation of mechanisms and algorithms. Conceptualisation tasks for predefined functionality are different as it is rather a configuration than actual code design that is required for language specific algorithms.

3.3 The Compilation Phase

The compilation phase is the last of the conceptualisation life cycle. In this phase it has to be decided in which way the realised hybrid modelling method should be offered to the end user. Depending on the scope and how comprehensive the method is in its composition some deployment alternatives are preferable to others, e.g. the deployment in form of a modelling app. The deployment variant also determines how the hybrid modelling method is intended to be used. The composition by the use of a terminal server and by means of a web-interface allows from the end user point of view quick access without taking care of any further installation routines. Furthermore the hybrid modelling method is more easily available for a bigger community. The deployment of the hybrid modelling method as an independent distributable unit, e.g. standalone version, is another option. Which deployment variant is meaningful depends on the actual 'business model' for the hybrid modelling method and represents the last design decision. In the compilation phase the actual realisation of the elaborated hybrid modelling is performed. We call this task customising as it is performed on an existing meta-modelling platform.

3.3.1 Platform Specifics

Depending on platform internal procedures, the actual sequence of implementation steps is predetermined. In order to allow defining the respective conceptualised parts, the provision of appropriate formalisms, e.g. programming or scripting languages, is required. These formalisms provide the actual support to codify the specifications and are either platform-specific, e.g. AdoScript for ADOxx® or rather generic like for example Java. ADOxx® offers integrated dialogs for the realisation of the conceptualised metamodel, the syntax creation in form of attributes and the design of the notation. Each dialog encapsulates specific functionality for achieving the expected result. For the mechanisms and algorithms 'message ports' are offered for a seamless integration. Depending on how specific the mechanisms and algorithms for the modelling technique are, not all implementation steps require actual coding. Some steps are solely configurations of generic platform functionality according to the hybrid modelling language needs, for example mechanisms for the notebook structure, predefining queries, print layouts or embedment of new menu buttons for launching algorithms and are therefore classified as 'customising' steps. The different nested formalisms guide the actual realisation on the platform and are prerequisite for a well-rounded, easy to handle modelling method.

3.3.2 Customising

The actual implementation belongs to the compilation phase of the conceptualisation life cycle.

In the ADOxx® meta^{2}model the central element is the 'library' that works as a container to which all formalisms and constructs of one of its instances, i.e. the modelling language metamodel are assigned to. The first step is the set-up of the platform-conceptualised metamodel of the modelling language. In this task, the ADOxx® meta^{2}model distinguishes between classes and relation classes. For relation classes at least two endpoints need to be defined and these are basically the specification *from* which modelling class *to* which modelling class the association is allowed to be drawn.

Once the basic 'skeleton' of the modelling language is given, the immediate next step would be the definition of the syntax. The modelling classes as well as relation classes have different attributes that are from predefined attribute types. For the platform the commonly known attribute types like integer, double, string, enumeration, etc. are defined. For the platform conceptualisation a distinction between class attributes and instance attributes is made. The difference between these two lies in the values the attribute can adopt. Class attributes are context neutral and not to be filled by the end user or modeller using the method once implemented. Instance attributes are context dependent and will be used by the modeller to capture data and convey certain information [8, p. 100].

In ADOxx® the concept 'notebooks', i.e. the dialog structure is defined by a number of attributes. 'Notebooks' need to be specified for those modelling and relation classes where attributes have been defined whilst the syntax realisation. In the ATTREP dialog of the platform, the attributes are summarised in chapters and sub-groups. Theses structural elements influence the display of the attributes for the 'end user' of the modelling method.

A further concept of the meta^{2}model platform shows, though on another level but most essential for the use of the modelling method, the structural element ‘model types’. A model type is a classification element for the available modelling concepts of a modelling method. For the end user model types are ‘a predefined set’ of modelling and relation classes that specify the purpose of a model. The concept of model types is similar to the ‘diagram types’ classification within the UML modelling language. The definition, which elements are within a model type and which are not, is sometimes difficult and can best be answered by ‘observing’ the actual use of modelling concepts, i.e. by means of the created models. The source of difficulties is actually if a model is created by means of a particular model type and the further development of that same model, i.e. same instance, should be done by modelling classes that are assigned to a different model type. Such a ‘model development’ procedure is related to the modelling procedure part but influences the implementation at this stage. A solution for avoiding such handling constraints is either to foresee the same modelling concepts in more than one model type or to work with ‘view mode’ concepts. The platform specifies in this context the ‘Mode’ concept.

The last part of the modelling language, the notation part is realised in the GRAPHREP dialog of the ADOxx® platform for the modelling and relation classes. As discussed, the notation can show static and dynamic parts whereas in this context the conceptualisation goes as far as to the description of the ‘behaviour’ of the notation, e.g. change of graphical representation if a certain condition is fulfilled. In the actual realisation it has to be determined if such a dynamic part is realised by the notation-specific formalism or if it is more efficient when realising it by means of a language-specific algorithm and hence a different formalism.

The actual coding of modelling technique-related algorithms forms together with the configuration of mechanisms the last customising part of a hybrid modelling method realisation.

Subsequently the compilation of the hybrid modelling method to a modelling tool winds-up conceptualisation life cycle.

4 Conclusions and Future Work

In the paper at hand it has been discussed that the design and conceptualisation of hybrid modelling methods require a number of steps where deliberate design decisions are of need. The conceptualisation is a prerequisite for an implementation in form of a ‘self-contained modelling tool’. In order to structure the different steps that are required for achieving this goal, a conceptualisation life cycle is suggested. Each of the single phases of the cycle shows a particular focus and contains parts where due to additional parameters provided by the platform a further design is of need. As the platform the hybrid modelling method is supposed to be realised on, provides functionality and structures which have not been relevant during the development phase of the hybrid modelling method ‘on paper’ more design decisions given by the platform logics are necessary.

Further work is required for integrating formalisms to describe language-specific algorithms apart from pseudo code which we consider as insufficient, providing mechanisms for reusability of concepts and transformations on a meta-model level.

References

1. BOC_Group: BPMS Method Manual for ADONIS 5.0, p. 1206. BOC Asset Management GmbH, Vienna (2012)
2. Brodie, M.L.: John Mylopoulos: Sewing Seeds of Conceptual Modelling. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 1–9. Springer, Heidelberg (2009)
3. Fill, H.-G., Redmond, T., Karagiannis, D.: FDMM: A Formalism for Describing ADOxx Meta Models and Models. In: Maciaszek, L., et al. (eds.) Proceedings of ICEIS 2012 - 14th International Conference on Enterprise Information Systems, Wrocław, Poland, Vol. **3**, pp. 133–144. SciTePress (2012)
4. Frank, U.: Multiperspective enterprise modelling: theoretical background and design of an object-oriented development environment (German original: Multiperspektivische Unternehmensmodellierung: Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung). Habilitation Thesis, Munich, R. Oldenbourg Verlag (1994)
5. 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)
6. Karagiannis, D., Visic, N.: Next Generation of Modelling Platforms. In: Grabis, J., Kirikova, M. (eds.) BIR 2011. LNBIP, vol. 90, pp. 19–28. Springer, Heidelberg (2011)
7. Kühn, H.: The ADOxx® Metamodelling Platform. In: 1st Open Models Workshop Methods as Plug-Ins for Meta-Modelling. http://www.openmodels.at/web/omi/blogs/-/blogs/1st-international-workshop-on-omi?_33_redirect=%2Fweb%2Fomi%2Fblogs (2010). Accessed 07 November 2012
8. Kühn, H.: Method integration in business engineering (German Original: Methodenintegration im Business Engineering. Unpublished Thesis p. 284. Faculty of Business Administration and Informatics, University of Vienna, Vienna (2004)
9. Mahr, B.: Information science and the logic of models. Softw. Syst. Model. **8**(3), 365–383 (2009)
10. OED: hybrid, adj. Oxford_English_Dictionary (ed.). Online version of September 2012. http://www.oed.com/view/Entry/89809?redirectedFrom=hybrid#eid (2012). Accessed 29 October 2012
11. Staud, J. L.: Enterprise Modelling (German original: Unternehmensmodellierung), p. 379. Springer, Heidelberg (2010)
12. Strahringer, S.: Metamodel (German original: Metamodell). Enzyklopaedie der Wirtschaftsinformatik Kurbel, K., et al. (eds.), http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/konzeptuelle-modellierung-von-is/metamodell/index.html (2012). Accessed 07 November 2012
13. i* Wiki.: iStar Tools. http://istar.rwth-aachen.de/tiki-index.php?page=i%2A%20Tools (2011). Accessed 20 April 2011

14. Xu, T., Ma, W., Liu, L., Karagiannis, D.: Synthesizing Enterprise Strategic Model and Business Processes in Active-i*. In: 2010 14th IEEE Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp. 345–354 (2010)
15. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: Social Modeling for Requirements Engineering, p. 760. MIT Press, Cambridge (2011)

Requirements Engineering for Emergent Application Software

Pericles Loucopoulos(✉)

Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece
p.loucopoulos@hua.gr

Abstract. The field of Requirements Engineering is arguably one of the most sensitive areas in the development of not only software but more importantly in the development of systems and organisational structures and processes supported by such systems. As service systems play an increasingly important role in today's economy, the ability of software to respond to emergent real-world contexts becomes a key-enabling factor to developing new and unpredictable business models. This paper, which is partly based on the keynote lecture given by the author at ICEIS 2012, considers the field of emerged enterprise application software and critically examines the applicability of the methodology factors underpinning much of the practice in Requirements Engineering, to such systems.

Keywords: Requirements engineering · Emergent software · Service systems · Teleological methodology

1 Introduction

The last decade has witnessed the rising significance of services in world economies [1] to the extent that in some countries services account for over 65 % of their GDP. According to Eurostat "an essential source of innovation, knowledge-intensive services (KIS) and even more so, high-tech KIS, are often seen as major engines of growth in modern economies." In particular, "the value added by the EU's high-tech KIS sector increased at an average annual growth rate of 6.6 % between 2000 and 2005". Although high-tech KIS comprise more than just software, it is fundamentally software that provides the necessary added value to these systems. Therefore, the effective and efficient development of such systems can have a major impact on the economic value of KIS.

The aim for this paper is to examine the role of Requirements Engineering (RE) in contemporary software development in the context of *emergent application software*. Emergent application software is regarded as a key component in the service industry of tomorrow [2]. Some key challenges outlined in [2] include:

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 18–28, 2013.
DOI: 10.1007/978-3-642-40654-6_2,

- Providing dynamically adaptable services for new innovative business models from different service providers.
- Adaptable services to be compliant with Service Level Agreements (SLAs). In general dealing with non-functional requirements (NFRs) such as quality of service, security and privacy.
- Extending cloud computing to allow for semantically connecting software services across enterprise domains.
- Lining existing services efficiently towards new business processes.
- Dealing with evolvable and adaptable requirements and in general dealing with these at the user side.

In examining the role of RE in this changing landscape of software development and infrastructure systems it is worth looking at the contribution of RE to date. The past 40 years or so have witnessed the emergence of a variety of techniques for helping capture, represent, share, analyze, negotiate, and prioritize requirements. This is evidenced by the volume and impact of a plethora of requirements-related papers published in related journals and conferences [3]. At the same time the practical nature of RE has meant that some of this research has influenced the practice primarily in areas such as business and system modeling [4]. At the same time we have witnessed a number of initiatives that attempt to support the RE lifecycle, initiatives such as risk-driven methodologies [5], requirements tracing [6, 7], model re-use [8], use of scenarios [9], use of visualization [10], use of business rules [11, 12], enterprise modeling [13] and goal modeling [14, 15] to name but a few.

A recent study of practices [16] and workshops involving both academics and practitioners, the results of which were reported in [17, 18], revealed four challenges: (i) the intertwining of requirements and contexts, (ii) the continuing evolution of requirements, (iii) the influence of architectural considerations and (iv) coping with complexity. These challenges apply across the spectrum of application domains as well as across different development paradigms.

There is a realization that, as Software Engineering (SE) capabilities have grown, so have aspirations and expectations for future systems and these largely outstrip capabilities. Brooks' [19] statement more than 30 years ago, that "the complexity of software is an essential property, not an accident one" remains true to this day [20]. It could be argued that there is a whole set of new advanced design requirements [17] that demand now more than ever close attention to the effectiveness of the SE process. The field's focus and scope has shifted from engineering of individual systems and components towards the generation, adaptation and maintenance of software-intensive ecosystems consisting of software, hardware, human and organizational agents, business processes and more. Such software ecosystems require attention in three areas namely those of design and evolution, orchestration and control and monitoring and assessment [21].

This paper examines the traditional approach to RE and argues that the prescriptive way of using maps of the development process is not valid for software that needs to exhibit emergent properties (Sect. 2). The need to be adopt a user-driven approach for deliberating requirements for emergent software is considered in Sect. 3. Section 4 argues that an intention-driven approach provides the necessary abstraction and flexibility that is required during the development and evolution of emergent software. The paper concludes in Sect. 5 with a reflection on different modeling dimensions that would support a development paradigm for emergent application software.

2 RE in the Context of Emergent System Development

The demand for emergent application software is due to the realization that software needs to adapt itself dynamically to support the ever-changing requirements of markets and enterprises. Enterprises making use of such software would explore this type of software to develop new business models that were not originally planned. Emergent application software is characterized by the combination of service components from a variety of providers in a dynamic and flexible manner. Let us examine the role of RE in attempting to design such systems. Is the traditional approach of using "maps of designs" (i.e. methods that follow a prescribed route from problem to solution) appropriate?

Developing emergent software falls in the category of design problems often referred to as ill-defined problems, or as Rittel and Webber called them "wicked problems" [22]. They demand that designers not only keep in pace with the technology's potential, but that they take business goals, social and human factors into account as well. In such a context, problem setting is no more a trivial activity. On the contrary, problem setting is, in a large part, intertwined with problem solving [23]. In ill-defined problems, defining the problem amounts to a large extent to solving it. Ill-defined problems are usually hard to solve because their definition, and not only their solution, is difficult. Usually there is no single definition, and subsequently no any single solution to be found. Ill-defined problems cannot be *optimized*; they can only be *satisficed*.

This point was argued by Fred Brooks in his talk at the workshop in "Design Requirements" [17], as follows: "*A point I want to emphasize in the requirements process is we do not usually know what the goal is. I will assert that this is a deep fact of reality that is ignored in much of the literature about requirements. We do not know what we are trying to build. The hardest part of most designers of complex systems is not knowing how to design it, but what it is you are trying to design. When we talk about eliciting requirements, we are talking about deciding what it is we are trying to design*". In other words the development tree emerges as we progress through designing and the development tree is not about decisions but about designs [20].

The traditional approach to designing is that of 'design by drawing'. In other words, an artifact is represented in terms of expressions of mental models. For example, a design drawing in the case of the architect may be the design of a building; or in the case of an information systems designer, the collection of conceptual schemata representing the form of the information system. This is an approach that concentrates entirely on the artifact itself. It attempts to visualize the end product prior to its implementation. However, a major disadvantage with this approach is that it ignores completely those issues that cannot be visualized, for example, social issues, issues of value conflict, etc. It is only by making the process open to inspection and critical evaluation that it is possible to model and therefore reason about such issues.

The expressions of mental models (the product of the design process) concerning the artifact are normally considered within a framework that *maps* the design process. These maps, expressed as different design activities, are advocated by method engineers as a way of planning the designer's work. On examination of these maps, it becomes obvious that rather than considering a description of the process, the maps are really a description of the products. A design map, let us take as an example the steps in a structured method, tells us what the designer should consider (e.g., data flows, stores, modules etc.) and what they should produce (e.g., a set of hierarchically organized structure charts) rather than how a designer actually works.

Design maps tend to be general and prescriptive and they are an attempt of method engineers to provide some ordering in the actions of the designer. In fact there is little evidence that design maps bear any resemblance to the way designers work. Although the field of Information Systems is rather poor in empirical work in this area, there is fortunately a considerable wealth of material from other disciplines that demonstrate that maps fail because problem analysis and problem solution are more closely knitted than usually thought, and this is what makes a problem ill-defined.

For example, Lawson [24] observed the design processes followed by different groups of people concluding that there was no generic pattern that was followed and much was dependent on the culture and background of the different groups. Similarly, Eastman [25] recorded the workings of experienced designers. The protocols revealed how the designers explored the problem as well as their attempts to a solution concluding that many requirements were not thought out in the abstract nor were they defined in advance of searching for solutions but rather they were discovered as a result of experimenting with different solutions.

One of their characteristics of ill-defined problems is that they have no definite formulation. Formulating them amounts, to a great deal, to solving them.

What are the implications of this? Mainly a realization that designing does not proceed in a well founded route from problem setting to problem solving but there is a continuous interaction between the two. This can be seen in a more appropriate model of the design process, that of 'generator-conjecture- analysis' [26] which postulates that designers first define what might be an important aspect of the problem, then they develop a tentative design on the basis of this and they subsequently examine it to see what else can be discovered about the problem. Analysis guides design and design guides analysis -and all in an effort to gain an understanding of the problem, of the situation at hand. In this 'trade off" situation, models and modelling play a crucial role. Models are not just outputs of the process but also inputs to the thought process.

Whilst the discussion thus far has focused on the process of development one has to be cognizant of the changes that are taking part partly driven by technology and partly by business and economic factors. Table 1 compares and contrasts some of the key differences between traditional and contemporary development settings.

Traditionally development of software-intensive systems assumed a reasonably stable business environment. Rapid market changes such as electronic commerce, deregulation, mergers, globalisation and increased competition have led to a business environment that is constantly and rapidly evolving.

Table 1. Traditional vs. Contemporary Development.

Traditional	Contemporary
The context is a reasonably stable system ecology	The context is a rapidly changing system ecology
Emphasis on business process improvement via IS	Emphasis on enterprise and market transformation via IS
A key issue is that of alignment	A key issue is that of innovation
System properties predictable	Emergent system properties
Development based on a decision paradigm	Development based on a design paradigm
Clear separation between system and user	The human is no longer outside the system but an integral part of it

This leads to the demands for exploiting information systems technologies not just for supporting business processes but also and perhaps more crucially in enabling organisations to transform their businesses.

This implies that nowadays there is less emphasis in carefully aligning a system to the business and instead seeking to innovate through the use of technology.

Whilst the properties of systems were traditionally predictable, there is a realization that systems need to adapt to their environment. This desire for systems with emergent properties raises issues with respect to requirements evolution, requirements tracing and requirements quality.

This uncertainty at the outset leads one to conclude that the decision-oriented approach i.e. one that has a pre-determined set of development actions is not applicable and a more design-oriented approach is required, an approach that affords, experimentation and re-work.

Finally, the user is nowadays an integral part of a system and is no longer a mere observer. This in turn raises demands for user-driven paradigms that enable users to define, design and review potential solutions.

3 RE for Emergent Software Services

As introduced already in Sect. 2, emergent application software is characterized by the combination of service components from a variety of providers thus exploiting Software as a Service (SaaS) as a software development and delivery approach. The salient feature of the SaaS approach is its flexibility, as it enables organizations to create new software applications dynamically to meet rapidly changing business needs. The SaaS approach has become a common software delivery model for many business applications, including accounting, customer relationship management (CRM), enterprise resource planning (ERP), invoicing, human resource management (HRM), content management (CM) and service desk management. The Open Group [27, 28], which is behind the development of Software Oriented Architecture (SOA) Reference model, defines SOA as "an architectural style that supports service orientation. Service orientation is a way of thinking in terms of services and service-based development and the outcomes of services".

With the advent of cloud computing, the SaaS approach has become part of the nomenclature of the cloud stack, along with IaaS (Infrastructure as a Service) and PaaS (Platform as a Service). Armbrust et al. [28] argue that what is really new in cloud computing is utility computing as a service, as cloud computing is the sum of SaaS and utility computing. These authors eschew the terms such as IaaS (Infrastructure as a Service) and PaaS (Platform as a Service) because they argue that accepted definitions for these terms still vary widely and the line between "low-level" infrastructure and a "higher-level" platform is not crisp. They believe the two are more alike than different, and hence consider them together under the general term of "utility computing".

This viewpoint suggests that the success of cloud computing depends on close collaboration between these users at all levels of the cloud continuum. The development of cloud computing should therefore have strong user engagement and involvement. Typical cloud users are SaaS providers, who, on one hand bring the benefits of cloud computing to end users – SaaS users, and on the other hand, drive the development of cloud applications through solving the end user problems. SaaS applications should therefore be able to scale seamlessly at all levels of the cloud continuum, from higher-levels that support the SaaS application development, all the way to the cloud platforms.

RE can play a very significant and central role in bridging the gap between technology and users by extending the abstraction level of SaaS towards the user end, which is conceptually depicted in Fig. 1.

A key factor in the scheme outlined in Fig. 1 is that of the "application independent model". These models would augment the typical way of SaaS working. SaaS focuses on delivering software on demand by enabling the composition of service applications, which is delivered at the point of need, executed and then discarded. Current efforts concentrate on the underlying technology and pay less attention to the ways that users may be engaged. Users may get engaged if the 'language' and the method deployed are closer to their perception. This would then enable users to develop and modify software without necessarily being information technology professionals. The application independent models would be archetypal models that would fall into three categories:

- *Component Models*. This type of model supports the development of component services. Examples are requirements models [29] and design models [30]. Component services are fine-grained service components that can be reused for composing different types of services. Component developers can use these models to match and select suitable component services offered by cloud providers or use them as a baseline for designing new component services.
- *Service Models*. This type of model supports the development of services. Examples are service descriptions, business rules [31], and quality of service specifications [32]. Services are higher-level, business-oriented components which are composed from one or more service components. Service developers can use these models to

match and select suitable services offered by cloud providers or use them as a baseline for designing new services.

- *Business Models.* This type of model supports the development of service systems or ecosystems as shown. Examples are business process models [33], service interaction and orchestration models. Service systems are business applications composed from one or more services. Developers can use these models to compose service systems on demand or use them to integrate service systems into an ecosystem.

There are many advantages in deploying the scheme shown in Fig. 1. Specifically:

- Services would be closer to user needs.
- Applications would be more adaptable to user needs.
- Implementation would be independent of any vendor.
- Implementation would be independent of any platform.
- Adaptivity would be easier for vendors.

Application independent models would represent a knowledge base to support the RE process, exploiting knowledge about specific domains [34, 35], developing and maintaining archetypical, generic solutions [36, 37], as well as assisting in the process of progressing from fuzzy expressions of requirements to formal specifications [38, 39].

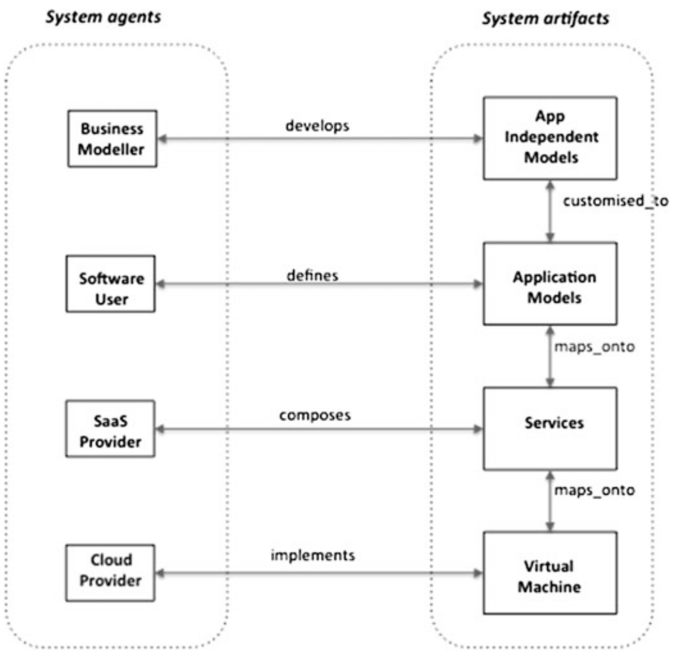


Fig. 1. Extending the level of abstraction towards the user-end.

4 Intention-Driven RE for Emergent Software Services

Design actions are cognitive actions, and are mirrored in the conceptual models employed by the designers. An important way of working in developing, managing and exploiting these conceptual models (for example, in Fig. 1 customizing the application independent models, to application models), is that of the teleological approach [15, 40]. The term teleological being used to convey the fact that *whatever* we do, we do to identify the *means* by which we shall be able to fulfill the *ends* stated (by means we refer to the lower level goals whose achievement is necessary -and, we hope, enough- for the high level ends). When we substitute a goal for a requirement, this goal is no more only a goal for the enterprise, the project, or whatever we might be working for; *it becomes a goal for us as well* (i.e. all those involved in the 'change' process).

The teleological approach advocates that everything done is incited by some goal. It is *not* a prescriptive way-of-working and in this sense it avoids the problems discussed in Sect. 2.

The task of someone using enterprise knowledge modelling is to determine the means by which an ultimate goal, let us call it G_0, will be realized. In attempting to achieve this, the process is governed by causal relationships between goals in a network of goals. That is, a directed edge from a goal G_i to another goal G_j implies that the achievement of G_i depends on the achievement of G_j. At every step of the process, the process is controlled or driven by the goal in hand.

The actions chosen for attaining the goals represent *working hypotheses*. As goals and sub-goals are established, these are tentative at least until they are tested about their fitness of purpose, i.e. the satisfiability of higher goals. This observation has its roots in the Principle of Rationality [41] which states that "if an agent has knowledge that one of its actions will lead to one of its goals then the agent will select that action". However, an agent may not have complete knowledge to make the appropriate action, or the set of actions may be so complex that it is impossible to determine the single correct set of actions that will lead to the achievement of the goal. In this more general sense, the Principle of Bounded Rationality [42] which states that "given a goal, an agent may not possess perfect or complete knowledge of, or be able to economically compute or access, the correct action (or sequence of actions) that will lead to the attainment of the goal" will apply. Therefore, any actions chosen by an agent in order to achieve a given goal will, in general, be a *hypothesis* that the actions will lead to the achievement of the goal.

We argue therefore that in developing emergent software systems, which cannot be prescribed at the outset, the teleological approach offers many advantages over other traditional techniques. The approach comprises both goals conceived as requirements and goals conceived as instances of personal efforts without making any assumptions about the sequence with which such goals are produced.

Using the teleological approach one is engaged into designing which itself constitutes the establishment of a series of causal relationships between goals. The result of this analysis could be visualised as goal graphs. These graphs therefore, would represent, a posteori, the design decisions showing successively refined situations that ultimately lead to the originally stated goal being satisfied. The goal graph will also show alternative design paths. It follows that the goal graph would by itself be a model of the design process. Or more accurately, the decisions adopted during the design process [43]. In this way it would be possible to trace decision from code to requirements, to observe and analyse the rationale used for certain design decisions and to achieve fast and effective changes to the software knowing the effects of these changes to the entire problem–solution frame.

5 Conclusions

This paper has put forward the position that the new class of emergent application software systems requires a user-driven paradigm, supported by a multi-perspective modelling approach. Such an approach would provide a convenient way of examining the different aspects that impact on the development of such systems. The three dimensions are those of: *product representation*, *change process* and *deliberation*.

The product representation dimension refers to the models that may be developed, browsed, manipulated etc. during the change process. The change process dimension refers to the steps, and activities made during the transition from a current situation to a future one. Finally the deliberation dimension refers to the participants, the arguments used, the choices made and the issues resolved during the transition process. The three dimensions are orthogonal in the sense that each aspect of one has a reflection on each one of the other two. For example, if we are engaged in discovering the existing situation (an aspect of change process) we may develop a business process model (an aspect of the product representation) and in doing so a number of stakeholders may engage into discussion and agreement (an aspect of the deliberation).

The task of emergent application software development should be viewed as a co-operative activity that exploits the contribution of different modelling views, each encompassing a specific type of knowledge. Within this multi-perspective approach enterprise analysis is based on two mechanisms: reasoning within a perspective; and reasoning across different perspectives in order to allow each individual step in the analysis process to exploit the most appropriate knowledge source.

References

1. Spohrer, J., Riecken, D.: Special section on service science. Comm. ACM **49**, 31–90 (2006)
2. ISTAG: Software Technolgies - The Missing Key Enabling Technology: Toward a Strategic Agenda for Software Technologies in Europe. ISTAG - Information Society Technologies Advisory Group, Brussels, Belgium (2012)
3. Cheng, B.H.C., Atlee, J.M.: Current and Future Research Directions in Requirements Engineering. In: Lyytinen, K., Loucopoulos, P., Mylopoulos, J., Robinson, B. (eds.) Design Requirements Engineering. LNBIP, vol. 14, pp. 11–43. Springer, Heidelberg (2009)
4. OMG (2003): UML Specifications. Object Management Group (2003)
5. Boehm, B.W.: A spiral model of software development and enhancement. Computer **21**, 61–72 (1988)

6. Gotel, O.C.Z., Finkelstein, C.W.: An analysis of the requirements traceability problem. In: Proceedings of the First International Conference on Requirements Engineering, pp. 94–101 (1994)
7. Ramesh, B., Jarke, M.: Toward reference models for requirements traceability. IEEE Trans. Software Eng. **27**, 58–93 (2001)
8. Rolland, C., Loucopoulos, P., Grosz, G., Nurcan, S.: A framework for generic patterns dedicated to the management of change in the electricity supply industry. In: International DEXA Conference (1998)
9. Kavakli, E., Loucopoulos, P., Filippidou, D.: Using Scenarios to Systematically Support Goal-Directed Elaboration for Information Systems Requirements. In: IEEE Symposium and Workshop on Engineering of Computer-Based Systems, pp. 308–314. IEEE Computer Society (1996)
10. Lalioti, V., Loucopoulos, P.: Visualisation of Conceptual Specifications. Information Systems **19**, 291–309 (1994)
11. Loucopoulos, P., Katsouli, E.: Modelling business rules in an office environment. SIGOIS Bulletin **13**(2), 28–37 (1992)
12. Tsalgatidou, A., Loucopoulos, P.: An object-oriented rule-based approach to the dynamic modelling of information systems. In: International Conference on Dynamic Modelling of Information Systems (1990)
13. Loucopoulos, P.: From information modelling to enterprise modelling. In: Brinkkemper, S., Lindencrona, E., Solvberg, A. (eds.) Information Systems Engineering: State of the Art and Research Themes, pp. 67–78. Springer (2000)
14. Kavakli, E., Loucopoulos, P.: Experiences with goal-oriented modelling of organisational change. IEEE Trans. Syst. Man and Cybern. - Part C **36**, 221–235 (2006)
15. Yu, E.S.K., Mylopoulos, J.: Why Goal Oriented Requirements Engineering. Department of Computer Science, University of Toronto (1998)
16. Hansen, S., Berente, N., Lyytinen, K.: Requirements in the 21st Century: Current Practice and Emerging Trends. In: Lyytinen, K., Loucopoulos, P., Mylopoulos, J., Robinson, B. (eds.) Design Requirements Engineering. LNBIP, vol. 14, pp. 44–87. Springer, Heidelberg (2009)
17. Lyytinen, K., Loucopoulos, P., Mylopoulos, J., Robinson, B. (eds.): Design Requirements Engineering. LNBIP, vol. 14. Springer, Heidelberg (2009)
18. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., Robinson, W.: The brave new world of design requirements. Information Systems **36**, 992–1008 (2011)
19. Brooks, F.J.: No silver bullet: essence and accidents of software engineering. IEEE Computer **20**, 10–19 (1987)
20. Brooks, F.P.: The Design of Design: Essays from a Computer Scientist. Addison-Wesley, New York (2010)
21. SEI: Ultra Large Scale Systems: The Software Challenge of the Future. Software Engineering Institute (2006)
22. Rittel, H.W.J., Webber, M.: Dilemmas in a General Theory of Planning. In: Cross, N. (ed.) Developments in Design Methodology, pp. 135–144. Wiley, Chichester (1984)
23. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J.: The Brave New World of Design Requirements: Four Key Principles. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 470–482. Springer, Heidelberg (2010)
24. Lawson, B.: How Designers Think: The Design Process Demystified. Butterworth, Cambridge (1990)
25. Eastman, C.M.: On the Analysis of Intuitive Design Processes. In: Moore, G.T. (ed.) Emerging Methods in Environmental Design and Planning. M.I.T Press, Cambridge (1970)

26. Hillier, B., Musgrove, J., O'Sullivan, P.: Knowledge and design. In: Cross, N. (ed.) Developments in Design Methodology, pp. 245–264. Wiley, New York, (1984)
27. OPEN-GROUP: SOA Reference Architecture. The Open Group http://www.opengroup.org/soa/source-book/soa/soa.htm (2009)
28. Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A.: A view of cloud computing. Commun. ACM **53**, 50–58 (2010)
29. Hagge, L., Lappe, K.: Patterns for the RE process. In: Proceedings of the 12th IEEE International Conference on Requirements Engineering, pp. 90–99 (2004)
30. Rapanotti, L., Hall, J.G., Jackson, M., Nuseibeh, B.: Architecture-driven problem decomposition. In: Proceedings of the 12th IEEE Internal Requirements Engineering Conference, pp. 80–89. IEEE Computer Society, Washington (2004)
31. Loucopoulos, P., Wan-Kadir, W.M.N.: BROOD: business rules-driven object oriented design. J. Database Manag. **19**, 41–73 (2008)
32. Heidari, F., Loucopoulos, P., Kedad, Z.: A Quality-Oriented Business Process Meta-Model. In: Barjis, J., Eldabi, T., Gupta, A. (eds.) EOMAS 2011. LNBIP, vol. 88, pp. 85–99. Springer, Heidelberg (2011)
33. Grefen, P., Ludwig, H., Dan, A., Angelov, S.: An analysis of web services support for dynamic business process outsourcing. Inf. Softw. Technol. **48**, 1115–1134 (2006)
34. Loucopoulos, P., Champion, R.E.M.: Knowledge-based support for requirements engineering. Inf. Softw. Technol. **31**, 123–135 (1989)
35. Loucopoulos, P., Champion, R.E.M.: Concept acquisition and analysis in requirements specifications. Software Engineering Journal **5**, 116–124 (1990)
36. Prekas, N., Loucopoulos, P., Rolland, C., Grosz, G., Semmak, F., Brash, D.: Developing Patterns as a Mechanism for Assisting the Management of Knowledge in the Context of Conducting Organisational Change. In: Bench-Capon, T.J.M., Soda, G., Tjoa, A.M. (eds.) DEXA 1999. LNCS, vol. 1677, pp. 110–122. Springer, Heidelberg (1999)
37. Seruca, I., Loucopoulos, P.: Towards a systematic approach to the capture of patterns within a business domain. The Journal of Systems & Software **67**, 1–18 (2003)
38. Loucopoulos, P.: The F^3 (From Fuzzy to Formal) View on Requirements Engineering. Ingénierie des Systèmes d'Information **2**, 639–655 (1995)
39. Al Balushi, T.H., Sampaio, P.R.F., Dabhi, D., Loucopoulos, P.: ElicitO: A Quality Ontology-Guided NFR Elicitation Tool. In: Sawyer, P., Heymans, P. (eds.) REFSQ 2007. LNCS, vol. 4542, pp. 306–319. Springer, Heidelberg (2007)
40. Loucopoulos, P., Kavakli, E.: Enterprise Modelling and the Teleological Approach to Requirements Engineering. Int. J. Intell. Coop. Info. Syst. **4**, 45–79 (1995)
41. Newell, A.: The knowledge level. Artif. Intell. **18**, 87–127 (1982)
42. Simon, H.A.: Models of Bounded Rationality. MIT Press, Cambridge (1982)
43. Louridas, P., Loucopoulos, P.: A generic model for reflective design. ACM Trans. Software Eng. Methodol. **9**, 199–237 (2000)

Blog Preservation: Current Challenges and a New Paradigm

Vangelis Banos[1(✉)], Nikos Baltas[2], and Yannis Manolopoulos[1]

[1]Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece
vbanos@gmail.com manolopo@csd.auth.gr
[2]Department of Computing, Imperial College, London, SW7 2AZ, UK
nb605@doc.ic.ac.uk

Abstract. Blogging is yet another popular and prominent application in the era of Web 2.0. According to recent measurements often considered as conservative, as of now worldwide there are more than 152 million blogs with content spanning over every aspect of life and science, necessitating long term blog preservation and knowledge management. In this work, we present a range of issues that arise when facing the task of blog preservation. We argue that current web archiving solutions are not able to capture the dynamic and continuously evolving nature of blogs, their network and social structure as well as the exchange of concepts and ideas that they foster. Furthermore, we provide directions and objectives that could be reached to realize robust digital preservation, management and dissemination facilities for blogs. Finally, we introduce the BlogForever EC funded project, its main motivation and findings towards widening the scope of blog preservation.

Keywords: Blogs · Blog preservation · Web archiving

1 Introduction

Blogs are types of websites regularly updated and intended for general public consumption. Their structure is defined as a series of pages in reverse chronological order. Blogs have become fairly established as an online communication and web publishing tool. The set of all blogs and their interconnections is referred to as the Blogosphere [1]. The importance and the influence of the blogosphere are constantly rising and have become the subject of modeling and research [13]. For instance, a 2006 study of the importance of blogs in politics, and for US Congress in particular, concluded that blogs play “an increasingly powerful role in framing ideas and issues for legislators and leaders directly” [27]. Blogpulse, a blog trend discovery service, identified 126 million blogs in 2009 and over 152 million blogs in 2010; while Tumblr, a relatively new blogging service, reports that they host over 33 million blogs [28]; statistics which undoubtedly prove the wide acceptance and dynamic evolution of weblogs. Moreover, they underline the importance of this novel electronic publication medium and exert its significance as part of contemporary culture.

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 29–51, 2013.
DOI: 10.1007/978-3-642-40654-6_3,

But despite the fast growth of blogosphere, there is still no effective solution for ubiquitous semantic weblog archiving, digital preservation, management and dissemination. Current weblog archiving tools and methods are ineffective and inconsistent, disregarding volatility and content correlation issues, while preservation methods for weblog data have not yet been duly considered. Indeed, existing Web Archiving solutions provide no means of preserving constantly changing content, like the content of weblogs.

Furthermore, to the best of our knowledge, no current Web Archiving effort has ever developed a strategy for effective preservation and meaningful usage of Social Media. The inter-dependence aspect of those media, demonstrated by weblogs featuring shared or adversary opinions, as well as weblogs that support, imitate or revolve around more central ones, is profoundly neglected. Two reasons are mainly responsible for this: firstly, the occasional harvesting of web resources and, secondly, their treatment as unstructured pages, leave little margin for capturing the aforementioned communication perspective of weblogs.

In this work, we present the new challenges that have to be met when facing blog preservation, including information integrity, data management, content dynamics and network analysis. Furthermore, we present the BlogForever EC funded project, its main motivation, objectives and findings towards widening the scope of blog preservation.

2 Related Work

Web preservation is defined as 'the capture, management and preservation of websites and web resources'. Web preservation must be a start-to finish activity, and it should encompass the entire lifecycle of the web resource [5]. The topic of web preservation was initially addressed in a large scale by the Internet Archive in 1996 [29]. Subsequently, many national memory institutions understood the value of web preservation and developed special activities towards this goal. Table 1 displays all major national and international web archiving projects which are part of the International Internet Preservation Consortium (IIPC).

As digital preservation techniques progress and awareness is raised on the matter, there is a continuous trend towards preserving more complex objects [7]. In the scope of web preservation, this means evolving from the preservation of simple web resources (i.e. html documents, images, audio and video files) towards preserving more complex web entities such as complete websites, dynamic web portals and social media. This trend is persisting with more social media content being considered for preservation. For instance, the Library of Congress has started preserving all Twitter content since 2010 [8].

The European Commission has identified the growing need to keep digital resources available and usable over time. To support research in the field, the FP7 ICT Research Programme 2009–2010 and 2011–2012 included specific provisions for digital preservation and web preservation under objectives ICT-2009.4.1: Digital Libraries and Digital Preservation and Objective ICT-2011.4.3 Digital Preservation [10]. A number of EC funded projects pursuing advanced web preservation are listed below:

Table 1. International internet preservation consortium members.

Organization	Years	Access Methods
Bibliotheca Alexandrina's Internet Archive, Egupt	1996	URL Search
Bibliothèque nationale de France - Archives de l'Internet	2002	URL Search, Keyword Search, Full-Text Search, Topical Collections
Government of Canada Web Archive	2005	URL Search, Keyword Search, Alphabetic Browsing, Full-Text Search
Croatian Web Archive (HAW)	2004	URL Search, Keyword Search
The Internet Archive (International)	1996	URL Search, Topical Collections
The Icelandic Web Archive	2004	URL Search
Finnish Web Archive	2006	URL Search, Full-Text Search
Kulturarw3 - The Web Archive of the National Library of Sweden	1997	URL Search
Library of Congress Web Archive, USA	2000	URL Search, Alphabetic Browsing, Subject Browsing, Topical Collections
Royal Library and the State and University Library, Aarhus, Denmark	2005	URL Search
Nettarkivet Norge (WebArchive Norway)	2001	Keyword Search
New Zealand Web Archive	1999	URL Search, Keyword Search, Alphabetic Browsing, Subject Browsing
National Library of Korea	2005	URL Search, Keyword Search, Subject Browsing
PANDORA Australia's Web Archive	1996	URL Search, Keyword Search, Alphabetic Browsing, Full-Text Search, Subject Browsing
Digital Heritage of Catalonia (PADICAT)	2005	URL Search, Keyword Search, Alphabetic Browsing, Subject Browsing, Topical Collections
Webarchive of Slovenia	2007	URL Search, Alphabetic Browsing
The UK Government Web Archive	1997	URL Search, Alphabetic Browsing
UK Web Archive	2005	URL Search, Alphabetic Browsing, Full-Text Search, Subject Browsing, Topical Collections
Web Archiving Project, Japan	2002	Keyword Search, Full-Text Search, Topical Collections
Web archive of The Netherlands	2007	URL Search, Keyword Search, Alphabetic Browsing, Full-Text Search, Topical Collections
WebArchiv - archive of the Czech web	2007	URL Search, Subject Browsing
Web Archive Switzerland	2008	URL Search, Keyword Search, Full-Text Search, Subject Browsing, Topical Collections
Webarchive Austria	2008	URL Search, Topical Collections

- **LiWA** (Living Web Archives) aimed to extend the current state of the art and develop the next generation of Web content capture, preservation, analysis, and enrichment services to improve fidelity, coherence, and interpretability of web archives [20].

- **ARCOMEM** (From Collect-All Archives to Community Memories) is about memory institutions like archives, museums and libraries in the age of the social web. Social media are becoming more and more pervasive in all areas of life. ARCOMEM's aim is to help to transform archives into collective memories that are more tightly integrated with their community of users and to exploit Web 2.0 and the wisdom of crowds to make web archiving a more selective and meaning-based process [11].
- **SCAPE** (Scalable Preservation Environments) project will address scalability of large-scale digital preservation workflows. The project aims to enhance the state of the art in three concrete and significant ways. First, it will develop infrastructure and tools for scalable preservation actions; second, it will provide a framework for automated, quality-assured preservation workflows; and, third, it will integrate these components with a policy-based preservation planning and watch system. These concrete project results will be driven by requirements from, and in turn validated within, three large-scale test beds from diverse application areas: web content, digital repositories, and research data sets [11].
- **LAWA** (Longitudinal Analytics of Web Archive Data) project will build an Internet-based experimental test bed for large-scale data analytics. Its focus is on developing a sustainable infra-structure, scalable methods, and easily usable software tools for aggregating, querying, and analyzing heterogeneous data at Internet scale. Particular emphasis will be given to longitudinal data analysis along the time dimension for Web data that has been crawled over extended time periods [18].
- **PATHS** (Personalised access to cultural heritage spaces) project goals are to provide innovative user-driven personalised access to cultural heritage collections and to support user's knowledge discovery and exploration. The project will create a system that acts as an interactive personalised tour guide through existing digital library collections by extending the state of the art in user-driven information access and by applying language technologies to analyse and enrich online content, with links to related items and background information.

The topic of web preservation in general and blog preservation in particular has been also addressed by a number of private startup companies throughout the world. Pagefreezer [24] is claiming to support web archiving and social media archiving. Another popular service is VaultPress [30], which provides security, backup and support for Wordpress blogs. Figure 1 displays a timeline of important web preservation projects and initiatives since 1995.

Despite the presented activities in the field of web preservation, we argue that there is still no effective solution for ubiquitous semantic weblog archiving, digital preservation and dissemination. Current web archiving tools and methods are not designed for the semantic web era and are ineffective and inconsistent, disregarding volatility and content correlation issues. Additionally, preservation methods for weblog have not yet been duly considered.

In the following section, we present a number of issues that arise when dealing with blog preservation and the current solutions to these issues.

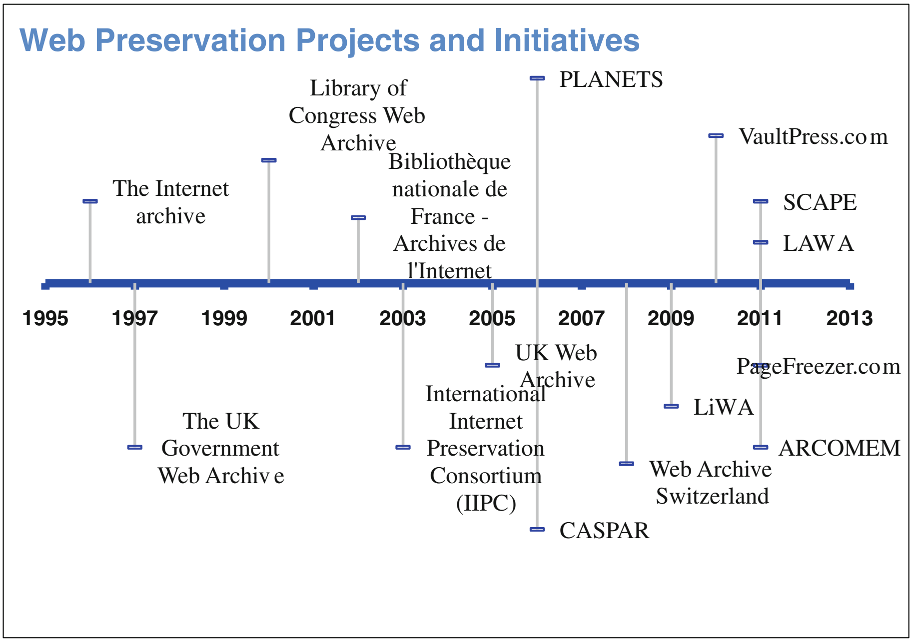


Fig. 1. Timeline of important web preservation projects and initiatives.

3 Blog Preservation Issues, Overview and Consideration

Blog preservation activities can be divided into three main groups: (a) content aggregation, (b) archiving, and (c) management. Here, we present the blocking issues for each one of these groups of activities.

3.1 Blog Content Aggregation

Existing web archiving solutions provide no means of aggregating and preserving constantly changing content, like the content of weblogs. The following two broad technical approaches are usually followed.

Firstly, there are initiatives that select and replicate web sites on an individual basis, an approach exemplified by the Web Capture Initiative [31] and by some projects developed by national archives. A second group of initiatives use crawler programs to automatically gather and store large sets of publicly available web sites. The Internet Archive follows this approach by taking periodic snapshots of the entire web since 1996. Other crawler-based initiatives have focused on national domains, e.g. the pioneering Swedish Royal Library's Kulturarw project, which is now discontinued [4]. A complete list of national web archiving projects is shown on Table 1. These initiatives are usually complemented by deposit approaches, where owners or administrators of websites choose to deposit the web content they are publishing to the repository.

Regardless of the target content, current initiatives employ general purpose web harvesting to collect their material. This approach, although easy to implement, results in problematic and incorrect web archives, especially for highly dynamic site types such as weblogs and wikis, which exhibit special characteristics. More precisely, current weblog content aggregation and digital preservation suffers from the following issues:

- **Web Content Aggregation Scheduling** is a common issue among web archiving projects, since all of them perform this task on regular intervals without considering web site updates. On the other hand, weblogs are extremely volatile and tend to be updated several times during the day, with new content from editors as well as user comments and discussions. As a result, a large amount of weblog content is not preserved, resulting in subsequent information loss and inconsistent web archives. For instance, Internet Archive's latest web preservation project, Archive-it [3], which uses the latest Heritrix web crawler [12], enables harvesting material from the Web as frequently as every 24 h, once per week, once per month, once per quarter, annually or just once. The same also applies to the popular [32], which is used by the Web Capture Initiative of the Library of Congress, the National Library of New Zealand and numerous other institutions worldwide.
- **Web Content Aggregation Performance** is also a major issue for current web preservation initiatives. Most projects use brute-force methods to crawl through a domain or a set of URLs, retrieving each page, extracting links and visiting each one of them recursively, according to a set of predefined rules. This process is performed periodically without taking into account whether the target site has been modified since the previous content aggregation or which components of the weblog have actually been updated. Unlike regular web sites, weblogs support smart content aggregation by notifying third party applications in the event of content submission or modification. Two technologies supporting this are Blog Ping [33] and PubSubHub [6]. Nevertheless, they are not utilized by current web preservation initiatives, resulting in a waste of computing resources.
- **Quality Assurance Checking** is performed manually or in a semi-automatic way for most web preservation projects. The widely used Web Curator Tool requires the administrator to perform a "Quality Review Task" while the PANDORA Archive's quality checking process [21] also requires human supervision.

3.2 Blog Content Preservation

Preservation refers to the long-term storage and access of digital or digitised content. Existing generic web archiving solutions suffer from several preservation-related shortcomings that render them as poor choices for weblog archiving. These relate to both the long-term storage of a weblog as well as to the access and usage of the preserved content.

1. Current web preservation initiatives are geared towards aggregating and preserving **files** and not **information entities**. For instance, the Internet Archive aggregates web pages and stores them into WARC files (ISO 28500:2009),

compressed files similar to zip which are assigned a unique identification number and stored in a distributed file system. Additionally, WARC supports some metadata such as provenance and HTTP protocol metadata. Implicit page elements, such as:

– Page title, headers, content, author information,
– Metadata such as Dublin Core elements,
– RSS feeds and other Semantic Web technologies such as Microformats [16] and Microdata [26] are completely ignored. This impacts greatly the way stored information is managed, reducing the utility of the archive and also hindering the creation of added-value services.

2. **Current Web Archiving Efforts** disregard the preservation of Social Networks and of interrelations between the archived content. However, weblog interdependencies demonstrated by the identification of central actors and peripheral weblogs, as well as by the meme-effect that applies to them, need to be preserved, to provide meaningful features to the weblog repository.
3. **Current Web Archive Scope is Limited** to monolithic regions, subjects or events. There is no generic web archiving solution capable to implement arbitrary subjects and topic hierarchies. For instance, the National Library of Catalonia has initiated a web crawling and access project aiming collect, process and provide permanent access to the entire cultural, scientific and general output of Catalonia in digital format [23].

   Alternatively, the Library of Congress has developed online collections for isolated historical events such as September 11, 2001 [19]. There is an ongoing debate, about benefits or disadvantages of one or another long-term preservation methodology. Many papers have been written and many conferences dedicated to this issue have appeared. It is surprising however, how little has been done at practical level.

3.3 Blog Archive Management

Regardless of the way a weblog is archived, current solutions do not provide users with meaningful management features of the stored information. For example, the Internet Archive stores weblogs as generic documents, listing one post after another, an approach that hinders if not forbids further weblog management. Examining the list of national web archiving initiatives (Table 1) one can see that out of 23 projects, only 8 support Full text search (34 %), 9 support Alphabetic Browsing (39 %) and 8 support Topical Collections (34 %). The most common feature available to all archives is URL Search.

Current solutions completely disregard the social aspect and interrelations of weblogs or other social media. Furthermore, due to the nature of periodic web crawling, users can only view the exact state of their weblog on prefixed dates or times. This solution cannot keep track of the evolving semantics and usage context of highly volatile hypertext pages like weblogs. For example, the Occasio News archive, which collects sites based on their relevance to social issues, only preserves specific snapshots from a certain newsgroup [22]. Articles do not follow a continuous timeline, a fact that renders their substantial analysis in the future impossible. This results in prolific loss of information with respect to recording the weblog's evolution.

Additionally, current weblog archives cannot preserve the information regarding how posts, relevance links or other weblogs affect the original content and how they led to its propagation or extinction. However, this process must be identified to be of high cultural and sociological value: it is essential to preserve the notions and reactions of contemporary society, the motivations and drives, the interactions between complementary and adversary approaches to certain topics.

Moreover, browsing the preserved Blogosphere through current Web Archiving solutions, like Internet Archive or PANDORA, remains a tentative if not impossible task. For example, within the framework of these solutions, weblog interrelations indicated in the form of Blogrolls are treated as regular hyperlinks of the retrieved Web page with no particular informational value. Not only does this approach lead to the risk of them being omitted during the harvesting stage, especially by domain specific web archives, but it also disregards the value of preserving how thematically correlated weblogs interact with each other.

Finally, though web archived content is generally classified into wide thematic, regional or temporal categories, there exists no robust categorization technique. Weblogs' topic metadata are omitted if they do not fall into the predefined categories. For example, inter-relational authorship information is rarely incorporated into the generic archive model. However, the authorship of electronic publication bears several interesting features, like identification of central actors with authority ranking, person searches and interrelations between authors and the role of anonymity. This has many channels of interest in text mining and the social networking and scientific communities, and would be a stronghold of web archives focusing on social network websites Moreover, the temporal aspect of each Web Archive merely relates to a specific web-snapshot acquired through harvesting. Our methods of real-time harvesting, result into a continuous observation of the lifecycle of a weblog and provide accurate representation for each weblog at any point in time.

As implied by the aforementioned facts, a large fraction of current weblogs lacks digital preservation or it is partially archived. Additionally, digital archives created by means of any of the above mentioned solutions do not guarantee correctness and consistency, thus preventing their effectiveness and their proper usage.

4 Directions Towards Robust and Effective Blog Preservation

In this section, we present our approach towards robust and effective blog preservation. This is a challenge that the BlogForever project (BlogForever) is addressing from four different perspectives: modelling, aggregation, preservation and dissemination. The project's objectives are presented and then each one of the perspectives is outlined.

4.1 Objectives

The project's strategic objective is to provide complete and robust digital preservation, management and dissemination facilities for weblogs. Towards this end, the following scientific and technological objectives have been identified.

Study Weblog Structure and Semantics. BlogForever aims to analyse weblog structure and semantics to understand the unique and complex characteristics of weblogs and develop a generic data model as well as an ontology-based representation of the domain. To achieve this, weblogs are required to be understood and managed in 6 aspects:

1. As physical phenomena,
2. As logical encodings,
3. As conceptual objects with meaning to humans,
4. As structural objects of networked discourse and collaboration for knowledge creation in large groups of humans,
5. As sets of essential elements that must be preserved to offer future users the essence of the object,
6. As ontologies created in a bottom-up manner by communities rather than specialists.

Additionally, weblog aggregation heuristics will be developed to allow us to determine the best practices for efficient data extraction from weblogs.

Define a Robust Digital Preservation Policy for Weblogs. Developing a robust digital preservation policy for weblogs is one of the key objectives. The policy will include the following information:
1. Preservation strategy considerations for assessing risk, requirements for accessing deposited content and long-term accessibility of digital objects, as these factors are deemed to have enduring value. Furthermore, the preservation approach is to be described, including actions that are considered necessary for immediate, intermediate, and long-term preservation. In terms of depositing, it is important to have structures that allow for easy retrieval (and this relates to extracting structures and mapping to them; but also to predicting what and how queries of the future will look like – depending on the amount of flexibility that is required, the data storing can be simpler, or more complex).
2. The Assessment of Interoperability Prospects, which intends to address collaboration issues with existing generic European Web Archiving solutions. Moreover, means for reliable content transfer from the digital archive to other digital repositories, in the event of project termination are to be proposed.
3. The Digital Rights Management Policy, which addresses weblog copyright issues and controls the access level for each item and user in the digital archive.

Implement a Weblog Digital Repository. BlogForever aims to implement a digital repository web application, which will collect, archive, manage and disseminate weblogs. The platform will have the following 2 main components:

Fig. 2. BlogForever platform architecture.

1. The weblog aggregation component, which will be capable of searching, harvesting and analysing large volumes of weblogs. The output of this component will be an information package encoded in XML which will be submitted to the digital repository component.
2. The digital repository component, which will be responsible for weblog data preservation. The digital repository will ensure weblog proliferation, safeguard their integrity, authenticity and long-term accessibility over time, and allow for better sharing and re-using of contained knowledge.

A detailed depiction of the BlogForever platform architecture is depicted in Fig. 2.

Implement Specific Case Studies. BlogForever aims to design and implement specific case studies to apply and test the created infrastructure on extensive and diverse sets of weblogs. The case studies will be both generic (collecting weblogs from a wide array of topics) and domain specific (for example, a case study in academic bloggers community). Thus the case studies will provide the required breadth and depth to validate the developed tools, and guarantee that the project's results could be successfully and widely replicated after the project ends. The impact of the digital repository will be also evaluated by monitoring system usage and gathering user feedback.

The case studies will begin in summer 2012 and are expected to be completed in August 2013. The largest case study will include 500.000 blogs.

4.2 Modelling

Working towards the objectives identified previously, we have already examined a number of tasks regarding modelling the blogosphere.

Weblog Survey. The BlogForever Weblog Survey report [2] outlines a principal investigation into:

1. the common practices of blogging and attitudes towards preservation of blogs;
2. the use of technologies, standards and tools within blogs; and finally,
3. recent theoretical and technological advances for analysing blogs and their networks.

This investigation aims to inform the development of preservation and dissemination solutions for blogs within the context of BlogForever. The objectives pursued in this study enabled discussion of:

- common weblog authoring practices;
- important aspects and types of blog data that should be preserved;
- the patterns in weblogs structure and data;
- the technology adopted by current blogs; and finally
- the developments and prospects for analysing blog networks and
- weblog dynamics.

To achieve the aims and objectives of this investigation, a set of review and evaluation exercises were conducted. The members of the BlogForever consortium jointly designed and implemented:

- an online survey involving 900 blog authors and readers;
- an evaluation of technologies and tools used in more than 200 thousand active blogs;
- a review of recent advances in theoretical and empirical research for analysing networks of blogs; and
- a review of empirical literature discussing dynamic aspects of blogs and blog posts.

Some key outcomes of the survey are the following:

- A large number of bloggers do not normally archive or preserve their work. Many of them, however, expressed willingness to deposit their blogs into archives.
- A large number of blogs were found to use a variety of media objects, but most of them used textual data. The use of photographs and moving images was also reported to be frequent. Nearly 90 % of all the blogs used self-created content, while 28.9 % used remixed data.
- The importance of rich media, links and citations was found to be important – having direct implications for blog preservation strategies.
- Blog users frequently relied on monitoring blog traffic, comments, subscriptions and feeds as measures of popularity. The use of ranking methods varied widely.
- Motivations for maintaining blogs were primarily personal – for sharing information and promoting discussion topics.
- When asked about the types of data that blog users would like to preserve in an archive, the majority expected their entire blogs, with posts and comments, to be preserved. Figure 3 illustrates the importance of preservation of all blog elements according to the BlogForever Survey.

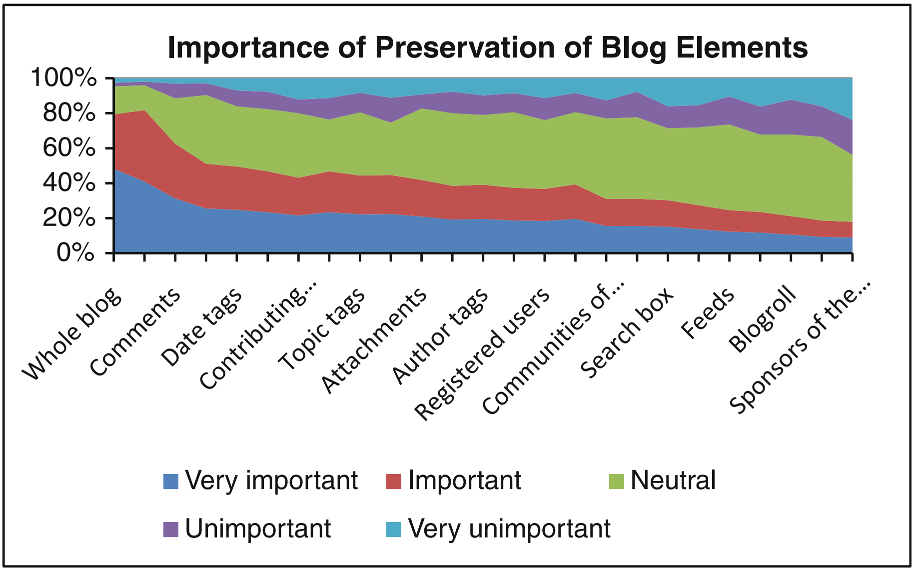


Fig. 3. Importance of preservation of blog elements according to the BlogForever Weblog Survey [2].

- Nearly 90 % of the authors interviewed never used an external service to preserve their blog and they mainly relied on their blog provider for these activities.
- Some hypotheses regarding the intention of blog authors to contribute their blogs to a central blog archive were tested the analysis shows that the perception of a collective benefit has a stronger influence as the perception of an individual benefit.
- These findings support the proposition that blogging is not seen by the authors as an individual activity even if the most blogs have only just one author. Instead it actually seems that bloggers are aware of the Blogosphere and intend to contribute to it.
- A detailed analysis of the survey can found in BlogForever D2.1 Survey Implementation Report [2].

Weblog Data Model. Our work on weblog data model [28] identifies the data structures considered necessary for preserving blogs by revisiting the earlier inquiry summarised in the BlogForever Weblog Survey. The report includes an inquiry into

- the existing conceptual models of blogs,
- the data models of Open Source blogging systems, and
- data types identified from an empirical study of web feeds.

The report progresses to propose a data model intended to enable preservation of blogs and their individual components. A generic blog data model representation displaying core components and their interconnections is shown in Fig. 4. This basic model can then be extended to ensure the integrity and authenticity of preserved blogs, satisfactory to the requirements of successful preservation and archiving.

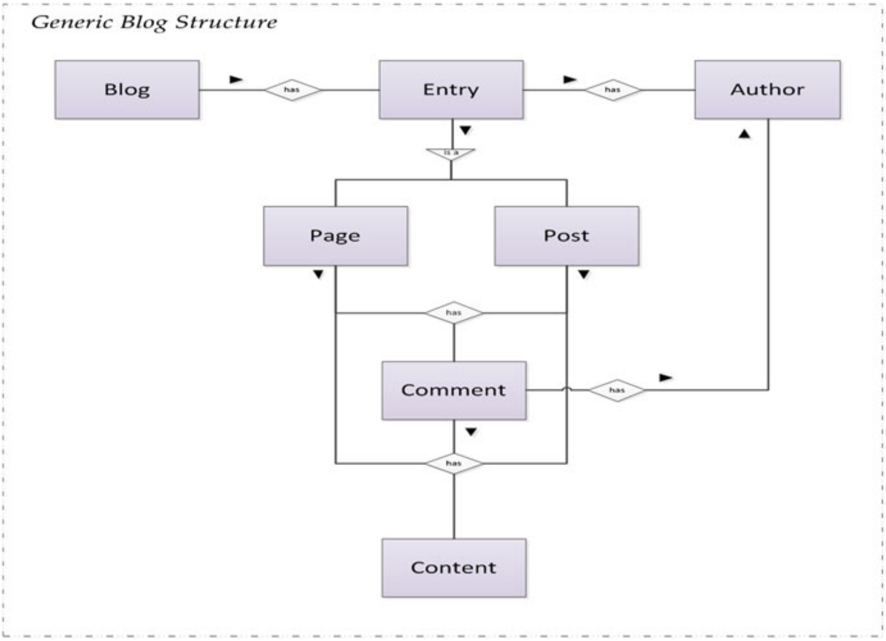


Fig. 4. Generic Blog data model elements and their relationships [28].

In addition to the core data model, a set of extension elements is also defined:

- **Weblog Context:** The entities described as part of the Blog Context component provide descriptive information about the blog and its elements in particular. It includes information about the selected presentation layer of the blog, description and keywords provided by the blogger, or the specific mark-up of individual elements of the blog.
- **Web Feed:** The Web Feed component consists of entities that are necessary to preserve information about web feeds of the blog.
- **Network and Linked Data:** This component contains the necessary associations that may exist across the blogs or their authors.
- **Community:** The Community component enables storing additional information about the active users – authors of posts and comments.
- **Categorised Content:** Categorised Content contains a number of entities that store the content collected from the blogs, which is decomposed into a number of smaller, but 'meaningful' pieces.
- **Standards and Ontology Mapping:** A mechanism for enabling the representation of stored blog data in specific standards, or for mapping it to certain ontologies.
- **Content Semantics:** They provide necessary structures to store the results of some analysis into the semantics of the content. For instance, the results of the sentiment analysis (i.e. sentiment scores) conducted on a specific piece of content can be stored along with additional data describing the algorithm, its version and the status of the results association with the content.

- **Spam Detection:** The Spam Detection component provides a mechanism for storing information about the algorithms and tools used for detecting spam and flagging the content included in the repository.
- **Crawling Information:** This component is intended to store information about the process of crawling. This will allow storage of information about the way crawling was conducted for a specific blog or sets of blogs. Storing information about crawling will make it possible to explain any differences between data along with the development of the crawler.
- **External Widgets:** External widgets make a fairly common appearance on blogs. Widgets are applications embedded into blogs or web pages. Some of the data describing the widget are planned to be stored as part of the preserved blog data.
- **Ranking Category and Similarity:** All of the entities described as part of this component are derived as a result of analysing captured blogs. These structures enable storing information about the ranking of blogs, or assigning them to certain categories.

A complete analysis of the BlogForever Data Model can be found in D2.2 Weblog Data Model Report [28].

Weblog Ontologies. Our work on weblog ontologies outlines an inquiry into the area of ontologies, conducted within the context of blog preservation, management and dissemination [15]. Three different scenarios regarding the application of ontologies are studied:

Semantic Extension of Tags. User generated tags and resulting folksonomies are widespread in blogs. However, while tags can organise blog posts inside a single blog according to the understanding of the blog author(s), it becomes more complicated if posts are aggregated from various blogs with possibly different contexts and topics. Therefore, it is necessary to identify and expose the meaning of the tags to overcome problems that result from the free choice of tags by different users, like homonyms and synonyms, and impair content retrieval.

Interoperability with Linked Open Data (LOD). Facilitation of the interoperability among repositories through the exposure and linking of data including explicit semantics. The application of ontologies can enhance the interoperability by the provision of open standards for describing, accessing, and connecting data. Figure 5 shows how interoperability could be established in this scenario.

The interoperability of the BlogForever environment has to be considered on two levels. First, there should be interoperability among different BlogForever archives. For example, a retrieval process for weblog data could operate on several archives and the results of complex search queries can be merged automatically. The use of shared vocabularies and a common ontology would allow an application to automatically merge the data from both repositories, providing a user of the repository with the means of searching and exploring the data as if they are from one repository.

Furthermore, interoperability with respect to other external repositories could be supported, for example, with other digital libraries. Digital libraries contain endless amounts of data that can be related to the data preserved in a BlogForever archive. Unlike interoperability between two BlogForever archives, the connection with

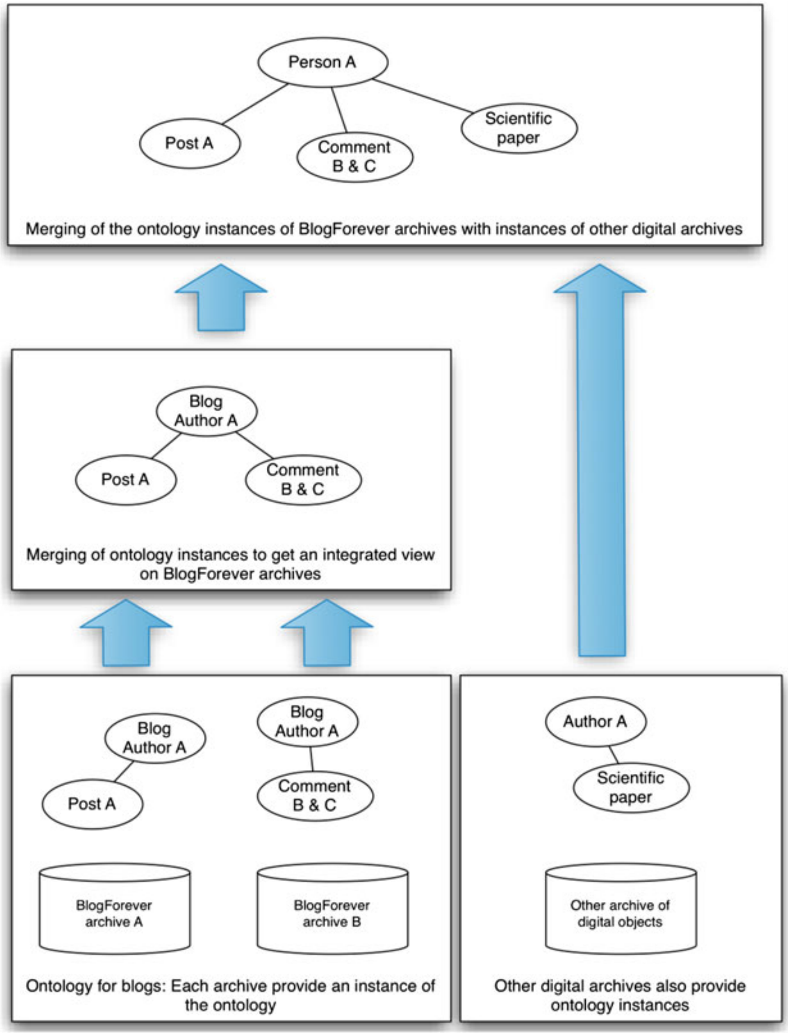


Fig. 5. Blog archive interoperability example using ontologies [15].

another digital library will extend the amount of concepts in the resulting ontology. In other words, two BlogForever archives share a common set of concepts (e.g. blog, post, blog author) and a merging means to merge instances of these concepts. However, another digital library has its own concepts like author, book, newspaper, etc. The relations between the concepts of both repositories have to be expressed (e.g. a blog author is a kind of author). Once the relations between the concepts are expressed formally, a merging of instances of both repositories will be possible.

Microformats, Microdata and RDFa. The utilisation of already available explicit semantics in the web pages that should be preserved can improve the quality of blog crawling. While data extraction on the layout specified by XHTML requires some heuristics to identify the meaning of data (e.g. to identify the author of a document), it can be obtained directly if microformats are available. Therefore, microformats, microdata, and RDFa will be examined regarding their possible utilization for data aggregation in BlogForever.

User Requirements and Platform Specifications. Requirements descriptions for the BlogForever platform were thoroughly investigated and assembled from several sources including already completed work; semi-structured interviews with relevant stakeholders and users' survey [14]. The report illustrates the method of interview conduction and qualitative analysis. It includes a description of relevant stakeholders and requirement categories.

The identified requirements were specified in a standardised template and modeled with the unified modeling language (UML). Thus, they can be easily explored and utilised by developers. Overall, the requirements are the foundation for the design phase because they represent the perspective of demand.

4.3 Aggregation

The first step to preserve blogs is to manage to achieve effective and complete blog content aggregation. This problem can be split down to two sub-problems, detecting blog updates and retrieving updated blog content.

Weblog Data Extraction Methodologies and Prototypes. The BlogForever Data Extraction Methodology Report [29] outlines an inquiry into the area of web data extraction, conducted within the context of blog preservation. In this work, we review theoretical advances and practical developments for implementing data extraction. The inquiry is extended through an experiment that demonstrates the effectiveness and feasibility of implementing some of the suggested approaches. More specifically, we look into an approach based on unsupervised machine learning that employs the RSS feeds and HTML representations of blogs. It outlines the possibilities of extracting semantics available in blogs and demonstrates the benefits of exploiting available standards such as microformats and microdata.

The detailed workflow presented in Fig. 6 represents the sequential process of data extraction and can be used to inform the design of the data extraction system of the BlogForever platform. The workflow branches depending on a number of conditional checks (depicted as diamonds). The first check looks for an available wrapper. The other branches determine the flows that enable both capturing and updating posts. The loops within the diagram illustrate the processes where more than one entry is subject to extraction. A complete analysis of this work can be found in BlogForever Report D2.6 Data Extraction Methodologies [29].

In addition to the development of a new data extraction methodology, a number of weblog data extraction prototypes were implemented to test the aforementioned techniques and evaluate alternative ways to implement the weblog spider component, one of the two key elements of the BlogForever platform. This work is continued in order to articulate an optimal set of weblog aggregation techniques.

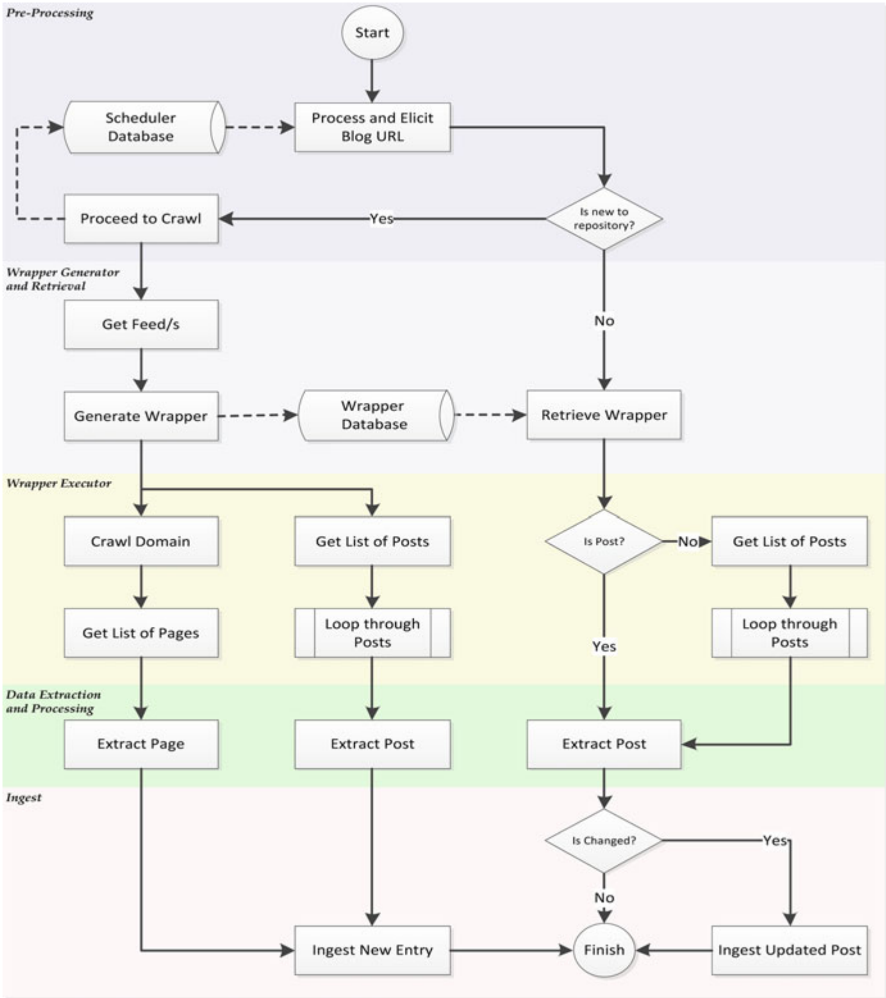


Fig. 6. Wrapper-based data extraction workflow (*Note: Dashed arcs indicate data flow*) [29].

Spam Filtering. Our research on spam filtering in the context of blog aggregation [17] comprises a survey of weblog spam technologies and approaches to their detection. While our work focused on identifying possible approaches to spam detection as a component within the BlogForever software, the discussion has been extended to include observations related to the historical, social and practical value of spam, and proposals of other ways of dealing with spam within the repository without necessarily removing them. We have identified three types of spam specific to blogs:

1. splogs, i.e. blogs that exist to promote affiliated websites, by influencing users to visit a webpage or buy a product, as well as spamdexing to undeservedly improve the ranking of a page in a web search by plagiarising content, stuffing keywords or creating large number of links,

2. blog comments that contain abusive content or are irrelevant to the original post, and,
3. fraudulent pings from non-blogs and/or splogs to attract visitors by misrepresenting content as fresh.

These spam types are analysed in the context of general web spam and various spam detection methods for weblogs are considered, concluding in a proposal for a spam detection workflow that might form the basis for the spam detection component of the BlogForever platform. The proposed methodologies in literature address web spam detection at the time of crawling, indexing and ranking. The types of features that can be used to detect spam can be grouped in spatial and temporal features. Spacial features refer to:

- **URL Pattern Information:** Methods using these type of features are based on the observation that:

  - spammers tend to stuff the URL with combination, mutations, and permutations of context rich keywords to benefit from the search engine ranking that rewards these URLs
  - spammers tend to use hyphens and long length URLs
  - domains that are cheap to acquire, such as ".info" domains, tend to be populated with a higher proportion of spam sites than expensive sites such as ".edu".
- **Home Page Content Information:** Methods using home page are based on the observation that:

  - spammers tend to repeat the same links and keywords
  - spam sites have short life span and grow very quickly
  - spammers employ a high percentage of nouns and only a few pronouns characteristic of expression of opinions
  - coherence of spam content may be lower that authentic content, that is the content would exhibit deviation from a general n-gram language model for $n > 1$
  - HTML templates created by automated blog creation software are repetitive
- **Feed-based Information:** Methods using feeds tend to copy home page content information techniques, and,
- **Link-based Information:** Methods using links are based on the observation that spammers try to exploit search engine ranking algorithms based on links, creating large numbers of fraudulent links.

Temporal features refer to the fact that the keywords, repetitiveness, network structure, size, and density with respect to splogs change at a different rate from that oberved with respect to authentic blogs.

To remedy the spam detection issues of blogs, a number of features and their pros and cons have been evaluated and are summarized in Table 2. Based on this information, we have formulated the proposed BlogForever spam filtering workflow which can be described in three main steps:

Table 2. Approaches to spam blog detection using various features.

Features	Pros	Cons
URL analyser/ template	Low process cost	Limited information – not very adaptive to change
IP/Post frequency	Could be difficult for spammers to manipulate	Must have history of updates and could become quite involved – e.g. where is the threshold for the frequency and how will it adapt to changes in the spam landscape?
Blacklist	Straightforward methodology and thrid party support available	Could lead to exploding blacklists.
RSS/content match	Indicates some level of agreement that the content is what the RSS feed says it is.	This requires that RSS feed is already available. Strictly speaking this is not spam filtering.
Full content analysis	Could be useful for removing duplicates. Difficult for spam to completely confound.	Could be process intensive.
User feedback	High precision	Labour intensive. Low recall because too many items for humans to examine.

1. Apply Ready-made Filters: black list databases (e.g. SplogSpot[1]), URL filtering, words in the content.
2. Apply Adaptive Filters: Ensemble classifier labels new posts, comments and blogs based on features such as temporal change, content, links.
3. Apply Adaptive Ranking Algorithms: Improve ranking performance of search engine by adopting positive crawling policy and implicit user feedback.

A complete analysis of this work can be found in BlogForever Report D2.5 Spam Filtering and Associated Methodologies [17].

4.4 Preservation

The process of digital preservation requires optimal retrieval and interpretation of the information to be preserved. As presented in the previous sections, our modelling and aggregation prototyping work will be the pillars upon which we will build an effective blog preservation platform.

Preservation Strategy. The preservation strategy will include information on assessing risk, requirements for accessing deposited content and long-term accessibility of digital objects, as these factors are deemed to have enduring value. Furthermore, the preservation approach is to be described, including actions that are

[1] http://www.splogspot.com

considered necessary for immediate, intermediate, and long-term preservation. In terms of depositing, it is important to have structures which allow for easy retrieval (and this relates to extracting structures and mapping to them; but also to predicting what and how queries of the future will look like – depending on the amount of flexibility that is required, the data storing can be simpler, or more complex).

Interoperability Strategy. Our planned work on the interoperability prospects of the BlogForever platform intends to analyse the different facets of interoperability: syntactic, semantic and pragmatic [25], by creating an interoperability testing methodology and specific scenarios. Furthermore, we are planning to address collaboration issues with existing platforms as well as libraries, archives, preservation initiatives and businesses that might be in synergistic relationships with BlogForever archives.

Digital Rights Management. Our planned work on Digital Rights Management (DRM) will initially include the identification and analysis of open issues and relevant discussions on the topic of blog preservation. Our aims will be protecting public access to information, content creators and content managers.

4.5 Management and Dissemination

To facilitate weblog digital preservation, management and dissemination, the project will implement a digital repository specially tailored to weblog needs. BlogForever digital repository will have to facilitate not only the weblog content but also the extended metadata and semantics of weblogs, which have been accumulated by the weblog aggregator as presented in Sect. 4.3.

The solution of creating a new software system as the basis of the weblogs repository has been considered and dismissed for this task, since many open-source repository back-ends are freely available on the Internet. In this respect, and taking into account the participation of CERN into the BlogForever consortium, the project will extend and adapt the globally acknowledged and widely used Invenio software [9]. The technology offered Invenio covers all aspects of digital library management. It complies with the Open Archives Initiative metadata harvesting protocol (OAI-PMH) and uses MARC 21 as its underlying bibliographic standard. Its flexibility and performance make it a comprehensive solution for the management of document repositories of large size and render it as an ideal basis for the BlogForever platform.

Long term blog preservation will be one aspect of the BlogForever platform. The other will be providing facilities for various stakeholders [14]:

- Content providers are people or organisations, which maintain one or more blogs and, hence, produce blog content that can or should be preserved in the archive.
- Individual blog authors are people that maintain their own blog.
- Organisations can serve as content providers if they maintain their own corporate blogs.
- Content retrievers are people or organisations which have an interest in the content stored in a blog archive and, therefore, they like to search, read, export, etc. that content.

- Individual blog readers are people who already read blogs for various reasons, e.g. family, hobbies, professional.
- In contrast, libraries operate more as a gatekeeper for individual retrievers. They provide access to various kinds of information sources, e.g. books, journals, movies, etc. Thereby, the access includes value added services like selecting and sorting the sources as well as adding metadata.
- Businesses also offer value added services based on the available information.

Each one of the aforementioned stakeholder has different blog preservation, archiving, management and dissemination requirements which have already been recorded and thoroughly documented, setting the priorities and work plan for the implementation of the BlogForever platform.

5 Conclusions

In this paper, we presented our perspective on the status of blog preservation and the blocking issues that arise when dealing with blog aggregation, preservation and management. Also, we identified a number of open issues that existing web archiving initiatives and platform face when dealing with blogs. Lastly, we presented an outline of the BlogForever EC funded project's current and future work towards creating a modern blog aggregation, preservation, management and dissemination platform.

Acknowledgements. The research leading to these results has received funding from the European Commission Framework Programme 7 (FP7), BlogForever project, grant agreement No.269963. We would also like to thank all BlogForever project partners for their invaluable contributions to the project.

References

1. Agarwal, N., Liu, H.: Blogosphere: research issues, tools and applications. ACM SIGKDD Explor. **10**(1), 18–31 (2008)
2. Arango-Docio, S., Sleeman, P., Kalb, H.: BlogForever: D2.1 survey implementation report. BlogForever WP2 Deliverable (2011)
3. Archive-it. Web Archiving Services. http://www.archive-it.org/. Accessed 11 April 2012
4. Arvidson, A.: Kulturarw3. In: Proceedings Conference on Strategies for the Internet: Preserving the Present for the Future, Copenhagen, pp. 101–104 (2001)
5. Ashley, K., Davis, R., Guy, M., Kelly, B., Pinsent, E., Farrell, S.: A guide to web preservation (2010)
6. Bhola, S., Strom, R., Bagchi, S., Zhao, Y.: Exactly-once delivery in a content-based publish-subscribe system. In: Proceedings International Conference on Dependable Systems and Networks (DNS), Washington, DC, pp. 7–16 (2002)
7. Billenness, C.: The future of the past – shaping new visions for EU-research in digital preservation. In: Proceedings Workshop, European Commission, Information Society and Media Directorate-General, Luxemburg (2011)

8. Campbell, L., Dulabahn, B.: Digital Preservation: the Twitter Archives and NDIIPP. In: Proceedings 7th International Conference Preservation of Digital Objects (iPRES), Vienna (2010)
9. CERN. Invenio. http://invenio-software.org/. Accessed 09 April 2012
10. Commission, European: Information and Communications Technologies (2011)
11. Edelstein O., Factor, M., King, R., Risse, T., Salant, E., Taylor, P.: Evolving domains, problems and solutions for long term. In: Proceedings 8th International Conference Preservation of Digital Objects (iPRES), Singapore (2011)
12. Heritrix.: IA Web Crawler. https://webarchive.jira.com/wiki/display/Heritrix/. (2012). Accessed 14 April 2012
13. Java A., Kolari P., Finin, T., Oates, T.: Modeling the spread of influence on the blogosphere. In: Proceedings 3rd WWW Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, Edinburgh (2006)
14. Kalb, H., Kasioumis, N., García Llopis, J., Postaci, S., Arango-Docio, S.: BlogForever: D4.1 User requirements and platform specifications report. Blogforever WP4 Deliverable (2011)
15. Kalb, H., Gkotsis, G, Pincent, E., Banos, V., Davis, R.: BlogForever D2.3 Weblog Ontologies Report. BlogForever WP2 Deliverable (2012)
16. Khare, R., Celik, T.: Microformats: a pragmatic path to the semantic web. In: Proceedings 15th International Conference on World Wide Web (WWW), Edinburgh, pp. 865–866 (2006)
17. Kim, Y., Ross, S.: BlogForever: D2.5 Weblog spam filtering report and associated methodology. BlogForever WP2 Report (2012)
18. LAWA. Longitudinal Analytics of Web Archive Data Project. http://www.lawa-project.eu/ (2012). Accessed 15 April 2012
19. Library of Congress. Web Archive. http://lcweb2.loc.gov/diglib/lcwa/html/sept11/. (2011). Accessed 10 April 2012
20. LiWA. Living Web Archives Project. http://liwa-project.eu. Accessed 15 April 2012
21. McPhillips, S.: PANDORA Archive technical details. http://pandora.nla.gov.au/pandoratech.html. (2012). Accessed 05 Aug 2004
22. Occasio News Archive Database. http://newsarchive.occasio.net/. (2012). Accessed 10 April 2012
23. PADICAT: The Digital Heritage of Catalonia. http://www.padicat.cat/. (2012). Accessed 10 April 2012
24. PageFreezer.com - Social Media and Website Archiving. http://pagefreezer.com. (2012). Accessed 10 April 2012
25. Papazoglou, M.P., Ribbers, P.M.A.: E-business: Organizational and Technical Foundations. Wiley, West Sussex (2006)
26. Rynning, M., Banos, V., Stepanyan, K., Joy, M., Gulliksen, M.: BlogForever: D2.4 Weblog spider prototype and associated methodology. BlogForever WP2 Deliverable (2011)
27. Sroka, T.N.: Understanding the Political Influence of Blogs: A Study of the Growing Importance of the Blogosphere in the US Congress. Institute for Politics, Democracy and the Internet. http://www.ipdi.org/UploadedFiles/PoliticalInfluenceofBlogs.pdf. (2006) Accessed 14 June 2009
28. Stepanyan, K., Gkotsis, G., Pincent, E., Banos, V., Davis, R.: BlogForever D2.6 Data extraction methodology report. BlogForever WP2 Deliverable (2012)
29. The Internet Archive. http://archive.org. (1996)
30. VaultPress - Safeguard your site. http://www.vaultpress.com. (2012). Accessed 10 April 2012

31. Web Archiving, Library of Congress. http://www.loc.gov/webarchiving/. (2012). Accessed 12 April 2012
32. Web Curator Tool Project. http://webcurator.sourceforge.net/. (2012). Accessed 12 April 2012
33. Winer, D.: Original announcement of blog ping. http://xmlrpc.scripting.com/weblogsCom.html. (2012). Accessed 12 April 2012

Part II

Databases and Information Systems Integration

Social Information Systems: Agility Without Chaos

Marc Quast[1(✉)], Mark J. Handel[2], Jean-Marie Favre[1], and Jacky Estublier[1]

[1]University of Grenoble, Grenoble, France
{marc.quast, jean-marie.favre, jacky.estublier@imag.fr}
[2]The Boeing Company, Seattle, WA, USA
mark.j.handel@boeing.com

Abstract. The state of the practice in large corporations shows that huge numbers of unofficial "shadow" applications are introduced by various actors to work around the limitations of the official information system. After a brief analysis of the present situation, this paper proposes an alternative enterprise architecture paradigm, reusing examples from a recent industry experience report to demonstrate how shadow application proliferation could be avoided without sacrificing business agility. We present our prototype implementation and its initial results, and discuss the possibility of applying consumer-space social mechanisms to both reduce the present chaos and leverage the collective intelligence and energy of the corporation towards cooperative, evolutionary development and maximum business agility.

Keywords: Information systems · Business applications · Enterprise architecture · Social software engineering · Software engineering paradigms

1 Introduction

Present enterprise information systems are a widespread source of frustration [1]. They are centered on heavy corporate applications, which cannot and indeed do not provide the agility required to survive in todays' competitive business landscape. In order to meet their objectives, business units cannot accept the poor service provided by corporate IT departments, and must build up independent IT resources to suit their specific or urgent requirements [2].

Due to the strategic importance of information systems, corporations typically don't volunteer evidence of their shortcomings. As a notable exception, The Boeing Company has recently published an experience report [3] describing the alarming numbers of *shadow applications* which various actors (business units, individuals, teams and communities) must introduce to work around the limitations of the official information system. Due to both our own experience in manufacturing industries[1] and

[1] The authors have a cumulated experience of over thirty years in the development and operation of business applications in industrial environments.

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 55–71, 2013.
DOI: 10.1007/978-3-642-40654-6_4,

discussions with professionals from other domains (banking, telecommunications, healthcare and government), we consider this a global phenomenon.

Though the benefit of "getting the job done" is sufficient to justify, and indeed pay for, their existence, shadow applications raise serious problems: duplicated and inconsistent data is commonplace, and having critical information and functionality scattered, unreachable and managed outside of standard IT processes is obviously not what comes to mind when envisioning a well-structured and robust information system. The result is a heavily fragmented application landscape, where consistency and governability have been sacrificed to achieve a reasonable level of business agility.

Our opinion is that with present information system architectures, no matter how carefully business applications are crafted, over time they will spawn shadow applications whenever resourceful actors have urgent unsatisfied needs, solving a local problem but leading to global chaos.

Our hypothesis is that a different business software architecture paradigm is both necessary and possible, and this paper proposes *social information systems* as such an alternative. After a description of the present situation, we propose an alternative architecture for business applications which could prevent the systematic recourse to shadow applications in their vicinity, using two use cases from [3] to illustrate its effects. We present our prototype implementation, and discuss the possibility of applying consumer-space social mechanisms to both reduce the present chaos and enable the cooperative design and evolution of a corporate information system.

2 Understanding the Present Chaos

The aforementioned Boeing experience report [3] provides the following compelling numbers of official, semi-official and unofficial applications for a big engineering organization.

In order to understand the present situation, the next sections provide a characterization of shadow applications, a description of our running examples and of the causes of shadow application emergence (Fig. 1).

2.1 A Characterization of Shadow Applications

Shadow applications are characterized by their purpose. If application B exists to work around the limitations of application A, or if B's features belong in A according to its

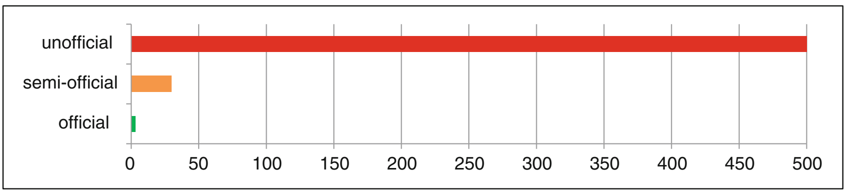


Fig. 1. Numbers of official and shadow applications from [3].

users, B can be considered a shadow application. This partial definition illustrates the subjective nature of the phenomenon.

Shadow applications are also characterized by their ownership. If it is owned by the IT department, it is an official application; otherwise it is a shadow application. The important distinction is not so much "IT or not IT" but "ownership by the actor effectively using the application". This allows the owner to quickly adapt the tool without consulting other parties or relying on the IT organization's priorities. It also provides him with full control over the visibility of the data and access to features.

For the purpose of our study, we will only retain the objective characteristic, i.e. ownership, and we thus define a shadow application as *"an application which is both functionally and technically owned by the organization using it"*.

Individual spreadsheets meet this definition. These are often used for simple data storage and manipulation, as a substitute for more robust business applications. This is a very common and possibly dominant use case since their introduction [4], and [3] qualify such spreadsheets as shadow applications. "Official" and "shadow" are relative concepts, and apply recursively at various levels of an organization. Multiple layers of shadow applications thus exist, the final one being *personal* applications. While shadow applications are typically loosely integrated with official or other shadow applications, manual synchronization is not uncommon [5].

Shadow applications are usually considered a "necessary evil" [5]. Organizations cannot work without them, but would prefer to avoid the data duplication they imply as well as the burden they represent in development and maintenance costs.

The benefits of shadow applications must clearly outweigh the drawbacks; otherwise line-of-business organizations would not develop, deploy, and maintain them. We will refer to the main benefits of shadow applications as perceived by their owners as the "**AVI** capabilities".

- The owner has full **A**utonomy to implement new features.
- The owner decides about **V**isibility of the application to the larger organization.
- The owner can **I**ntegrate (manually or automatically) with other applications.

2.2 Examples of Shadow Applications

In this paper, we will use fictional examples derived from the information disclosed in the aforementioned experience report [3].

- *"Luxury can report delays on process instances, but not the reasons for these delays which are managed by a shadow application."*
- *"Sometimes the tasks tracked by Luxury were informally decomposed into subtasks; (…) Luxury had no provisions for this kind of task decomposition."*

We make the assumption that Luxury tracks requests, a common use case in engineering environments. Figure 2 shows a fictional official application and two of its shadow applications, managing delay analyses and subtasks respectively.

While spreadsheets are arguably the most common form, shadow application architectures are limited only by the owner's resources, including full-blown business applications and, more fashionably, third-party applications in the "stealth cloud", i.e.

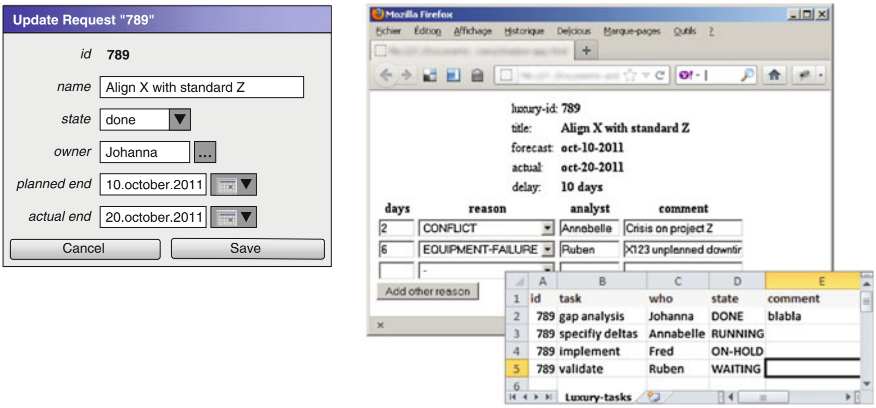


Fig. 2. On the left: Example of fictional official application. On the right: Examples of fictional associated shadow applications.

cloud services being consumed by business users without the knowledge, permission or support of the IT department [6].

2.3 Causes of Shadow Application Emergence

Shadow applications emerge to work around the shortcomings of official applications [2]. Thus we need to understand the causes for these problems.

Large organizations are not consistent and orderly systems. Referring both to groups and individuals, Kling [7] describes working relationships as *"multivalent with and mix elements of cooperation, conflict, conviviality, competition, collaboration, control, coercion, coordination and combat (the c-words)"*. Requirements from different stakeholders are thus often divergent or conflicting, which explains why the difficulty of requirements engineering increases exponentially with the number and diversity of participants. In [8], Ackerman indicates that when there are hidden or conflicting goals, people will resist articulating these. Under such circumstances, it is a challenge to converge on a consistent set of requirements and deliver a working application at all. But widespread dissatisfaction with the result is almost guaranteed by construction.

As an aggravating factor, corporations are not static. They must adapt to changes in their environment like new markets, technologies or regulations. Though the aforementioned c-words impact is often obvious at the time of application introduction, the continuous evolution of business requirements turns this into a subtle though continuous problem. Any change in any stakeholder's universe can invalidate the initial compromises and demand new rounds of discussion, yielding further dissatisfaction.

Besides inter-organization conflicts, some c-words foster shadow application emergence by themselves. A successful shadow application and the knowledge it captures is usually highly visible within an organization, and its ownership provides recognition (competition) and power (control, coercion).

There are other contributing factors. The widespread practice of reducing IT costs lowers both reactivity and quality of IT support, inciting business units to help themselves [9]. Technical obsolescence, a consequence of either respectable age or unfortunate choice of foundation technologies, can make it difficult to find the right skill set to implement changes. This paper focuses on the following factors leading to shadow application emergence.

- Business unit considers it impossible to converge on a single set of requirements fulfilling all stakeholders' requirements.
- Business unit does not want to rely on slow or expensive third parties.
- Business unit considers it in its best interest to produce a new system they own.

2.4 Preventing Shadow Application Proliferation

Our opinion is that with present software architectures, no matter how carefully official applications are crafted, over time they will spawn shadow applications whenever resourceful communities have urgent unsatisfied needs.

Our hypothesis is that if an application provides the AVI capabilities, the need for shadow applications is greatly reduced. Today's software architectures cannot provide these capabilities because the components of a business application (such as data elements, workflows, or forms) are *shared* among organizations. This sharing is both the main reason why business applications exist and the main reason for the emergence of their shadow counterparts. We therefore propose a new information system paradigm based on an architecture with a fundamentally different sharing principle.

3 Requirements for an Alternative Information System Paradigm

In this section we attempt to express the AVI capabilities (see Sect. 2.1) as a set of requirements, using the following definitions.

- An *Actor* is an *individual* or a *group* of individuals, for example the entire company, an organization, department, project team or community.
- *Elements* are runtime application components, like business entities (in our previous example a "*request*"), workflows, forms, reports, etc.
- *Applications* are collections of related Elements.
- An *Information System* is a collection of Applications.

3.1 Functional Requirements

Our first two requirements cover the most central operations in shadow application development.

R1: *Actors can extend existing Elements.*
R2: *Actors can add new Elements.*

Example 1 below reuses an observation from [3] to illustrate how an application satisfying R1 and R2 could defuse the need for shadow applications.

Example 1 – *Luxury*[2]
Official application "*Luxury*" manages *Request* entities, with among others attributes *title*, *state* and *delay*.

The "Quality" department needs to record the reasons for delays when they occur. Using R2, they introduce a new Element *DelayAnalysis* with attributes like *reasonForDelay* and *analyst* and associations with existing Element *Request.* Behind the scene, this leverages R1 to extend the *Request* Element with the reverse association *delayAnalyses*. This blends the new Element and extensions with the original *Luxury* entities thus enabling intuitive bi-directional navigation.

Other operations are adding missing attributes to an existing business element or adding more detailed states in an existing workflow. Example 1 highlights a new problem: the extensions are of interest only to a subset of the application's users, and may be confidential. To avoid cross-Actor pollution and conflicts, both R1 and R2 imply that Actors are *isolated* from each other by default, which yields the requirement R3.

R3: *Actors have private spaces.*

Elements are hosted in such private spaces and are by default not visible outside of them. We call these spaces *Perspectives*. In a typical enterprise setting, today's official applications would be Perspectives providing 'scaffolding' Elements, i.e. skeletons of business entities and associated high-level rules and functionality. Organizations at various levels would have their own Perspectives, hosting the extensions and additional Elements reflecting their concerns and level of detail. Individuals could likewise replace their spreadsheets with private extensions and Elements hosted in a private Perspective. However, completely isolated Perspectives would defeat the purpose of enterprise applications, which yields R4.

R4: *Actors can share the Elements they own.*

---

2 In the report, *Luxury* refers to both a business process and the supporting official application(s). We only refer to the latter here.

Perspectives can make selected Elements visible, either to everybody ("public") or to a restricted set of Perspectives. We call this operation *export*. Obviously the previously mentioned official Perspectives would export their Elements to all users. And business-unit-level Perspectives would export their Elements to the relevant Actors. Even individuals can share their Elements with others.

It is interesting to note here that R4 empowers all employees to contribute to the corporate information system, which we think provides significant benefits we will discuss later in this paper. The downside is that this could lead to cacophony through an overwhelming amount of available Elements, dictating R5.

R5: *Actors can select relevant Elements.*

A symmetrical *import* operation is therefore necessary. An Actor must be able to select, among all Elements available to him, only the ones he considers relevant. Instead of building his environment from scratch an Actor would *inherit* the Elements from the groups he belongs to; as a corollary, he must be able to *unimport* these Elements if they are not relevant for him. Example 2 below, again from [3], illustrates how R1-R5 could have avoided another real-life shadow application.

Example 2 – *Fallen*
Official application Fallen had produced a shadow application which added translations into Japanese next to English data fields.

Extending existing entities with additional attributes is a typical use case of R1. Such extensions would be owned by the Japanese branch of the company, and hosted on their servers in a Perspective (R3) we can call http://fallen.acme.co.jp/Translations.

All employees of the Japanese branch would inherit these extensions, and some groups or individuals could even choose to unimport the initial English attributes (R5). The extensions could also be exported to Japanese-spoken employees in other regions (R4).

Our previous use of the term Application encompassed a broad spectrum, from full-blown enterprise systems to private spreadsheets. Likewise, for Perspectives we envision a broad range from big Perspectives hosting self-sufficient third-party applications to tiny individual Perspectives with just a few extensions replacing spreadsheets. Some Perspectives may just factorize the optimal list of import and unimport declarations for a given organization or community.

3.2 Usability Requirements

A significant percentage of today's shadow applications are created by people without software development skills using office software like spreadsheets [4]. This observation makes usability a key requirement.

R6: *No programming is required for R1-R5.*

The last item may sound like reviving the dream of software without programmers. However, the data-centric nature of business applications makes it much less difficult for end-users to participate than more feature-centric software; significant contributions of entities, attributes, simple formulas and associations can be made through a forms-based interface, especially in the presence of example instances [10].

Contributions are not limited to what can be done by end-users through forms. A language-based representation of perspectives and elements is still necessary for professional software developers. Even for business units, contractors and interns have always been a means to get access to development skills beyond their internal competence to implement complex shadow applications. In a perspective-centric architecture, such expert contributions would still be possible, with the benefit of being better integrated with the rest of the information system.

4 Architecture

Figure 3 shows a possible high-level architecture meeting R1-R5. It iscentered on a classical enterprise directory component with *Users* and *Groups*. *Perspectives* are hosted by *Repositories*. Perspectives can define *Fragments*, which can be either self-sufficient (R2) or extensions of a Fragment from another Perspective (R1). Repositories can live on different servers.

At runtime, a User opens a *Session*, which determines a set of Perspectives – owned by the User or inherited from the Groups he is member of. This in turn determines a set of Fragments, which can be woven into *Elements*. The Session

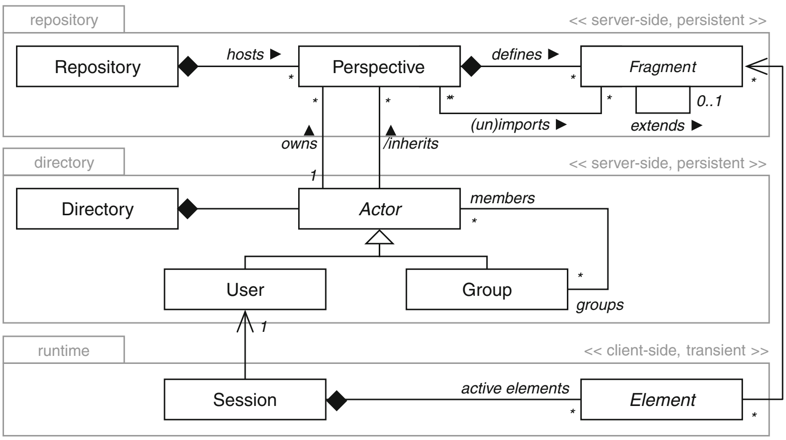


Fig. 3. Meta-model of Perspective-centric architecture.

becomes the Application, tailored to the connected user's profile. We call this a *virtual private application*, private because it reflects the user's unique combination of elements, virtual because it does not exist outside of the session.

In a perspective-centric architecture, applications are thus dynamically composed at runtime. Today, commercial-off-the-shelf (COTS) applications like issue-tracking, PLM and ERP systems need to suit the requirements of a variety of customers and provide some degree of flexibility through configuration and customization mechanisms [11]. We consider our proposal a generalization of these mechanisms.

5 Prototype

We have designed and implemented a first prototype of a perspective-centric system. Considering the complexity of the general case of extensible Elements, our prototype mainly focuses on data, i.e. business entities.

5.1 End-User Experience

Our main objective was to verify that the dynamic, perspective-centric nature of the system could be made transparent to end-users during normal use. The two screenshots on the left show two different users connected to a Luxury-like application, both displaying a request object. The first user belongs to the quality group and thus sees DelayAnalysis objects (1), the second user is from the planning group and sees SubTask objects (2) (Fig. 4).

It is important to stress the *additive* nature of the system, as opposed to subtractive, i.e. filtering. In a filtering approach, somewhere an Element would exist with all

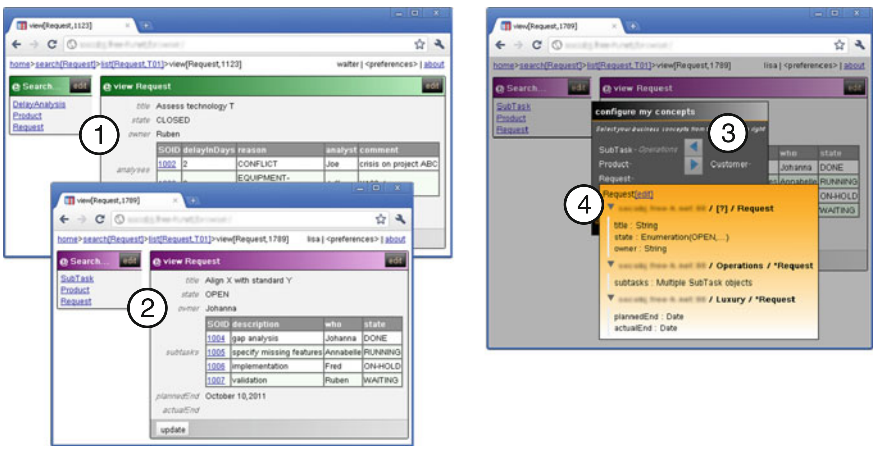


Fig. 4. On the left: two different users during normal use. On the right: a user inspecting his model.

attributes, which get filtered out depending on the users' profile. In our approach, Fragments exist in various places and get pulled together by the Session.

The main visible difference with a regular system is the presence of edit buttons, which allow inspection and tailoring of the connected user's model as illustrated in the screenshot on the right, which shows the possibility to import another Entity "Customer" (3), and that Element "Request" is a composition of Fragments from three different perspectives (4).

An ideal interface should have the intuitiveness of a spreadsheet, where filling an empty "header" cell transparently creates an extension with the new attribute, with default type and visibility. We believe the presence of actual records makes such example-centric modeling possible.

5.2 Implementation

It may appear natural to host extensions on the same server as their root Elements. However, to fully meet R1 and R2 any Actor must be able to provide his own storage and computing resources for extensions. Otherwise, although independent in functional terms, he is dependent from a physical resource point of view. This constraint dictates a *distributed* architecture, where Perspectives can be hosted on distinct servers and are pulled together at runtime by a client session.

It is important to guarantee that official systems cannot be disrupted or slowed down by extensions hosted on unreliable servers. No organization would accept an architecture with the potential for any unfortunate experiment by an employee to degrade access to central services. This constraint dictates *asynchronous* communication between components, allowing results from a high-reliability official system to be displayed without waiting for the extension results which may arrive later or never.

The prototype implementation is broken down in the following components.

- A central *directory* component, which in a real setting is the enterprise directory server where users and groups would just need to be annotated with references (URLs) to their associated perspectives.
- *Repository* components, which host Perspectives with entity definitions, extensions and associated instances, persisted in a database and exposed through web services.
- On the client-side, the *client session* component communicates with previous components to build a data model at runtime, and a *dynamic user interface* builds simple forms by inspecting this model.

Figure 5 illustrates the main interactions between the components, at initialization-time (1-3) and during regular use (4).

In step (1), the client authenticates the user and gets as a reply the full graph of his groups and perspectives. The client session then (2) requests all perspectives and the associated Fragment declarations from the various repositories involved. Receiving a Fragment triggers the (3) weaving mechanism which composes Elements. Usage is then similar to any distributed system, i.e. accessing an object triggers several requests (4).

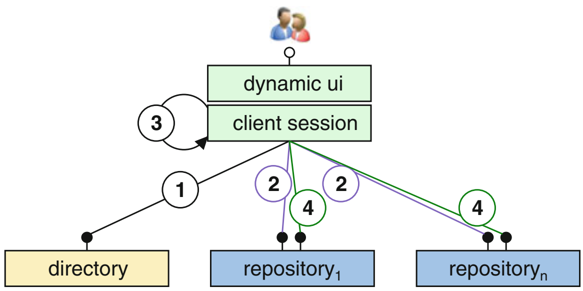


Fig. 5. Main interactions of the prototypes' components.

The communication between components is standard REST over HTTP. The protocol has been kept simple in order to enable integration of legacy systems in a perspective-centric landscape through the development of wrappers.

5.3 Evaluation

The main conceptual limitation of our first prototype is the focus on data only. Considering the centrality of data in business applications, we think that the results presented in the next section still represent a significant contribution.

From a technical point of view, the prototype has demonstrated the feasibility of asynchronous runtime composition of a data model, the transparency for end-users during normal use, and end-user update of the data model in their own perspective.

As a first proof-of-concept, we have instantiated the prototype with a project tracking use case and a configuration of 3 groups and 5 individuals with different perspectives. The prototype has been able to compose the individual models on the fly, proving the validity of the concepts of Perspective and Fragment.

We have presented this prototype to 8 information system professionals from 6 different industrial and educational organizations. All of them have over 20 years of experience and have witnessed the emergence of numerous shadow applications. Though they did raise some concerns, covered in the discussion section of this paper, their reactions to the proposal varied from fairly positive to enthusiastic. 4 out of 8 subjects have volunteered for evaluating the prototype with real application data.

As a second proof-of-concept, we have instantiated the prototype with the Luxury-like use-case presented in previous sections of this paper. The "Luxury" perspective and its associated "quality" and "operations" perspectives have allowed a unified representation of the three points of view. We were able to walk through use cases of both the Luxury and Fallen shadow applications, and show that technically they would have been avoided with a mature perspective-centric implementation.

The highly dynamic nature of the proposal initially made all interviewed professionals uncomfortable, illustrating the conservative attitude they adopt regarding the architecture of business applications, particularly the data layer. One manager has expressed a desire to restrict the perspective-centric nature of an application to the initial phases of its life, and to *"freeze"* the model once it has been collaboratively built and validated. This directly contradicted his earlier statements of continuously evolving and conflicting requirements, which he acknowledged.

We have measured the time required for the client session to compose one single business concept with 100 attributes. Without any optimization, the response time seems to be a linear function of the number of perspectives, with absolute values on a low-end laptop between 0.13 s for 10 perspectives (10 fragments with 10 attributes each) and 2.5 s for 100 perspectives (100 fragments with 1 attribute each). Compared to a user juggling with multiple shadow applications, we don't consider this overhead significant.

Further evaluation is required to validate our initial results, which represents a challenge as we will discuss in the next section.

6 Discussion

Perspectives represent different, finer and more connected information system grains than applications. We think they allow an information system to evolve organically in a unified and more controlled way than today's proliferation of shadow applications, without sacrificing the business units' ownership of their specific application elements. The reactivity and autonomy their mission demands is thus preserved.

6.1 Towards Social Information Systems

A perspective-centric application architecture represents a major shift of responsibilities from IT departments towards the community of users, not unlike the freedom spreadsheets have provided [4]. An IT department's main responsibility would be to provide the platform on which anyone (the IT department itself, but also business organizations and individual employees) could contribute elements in their area of expertise. We think this could leverage the collective intelligence [12] and energy of employees to collaboratively build and maintain the corporate information system, in a form of internal crowd-sourcing.

Considering today's mostly feudal management of information systems, this is a fairly disruptive proposal. Indeed, during our interviews most subjects have initially raised the concern that it could *"result in chaos"*. This reaction typically takes the official applications as a reference, which in reality only represents the tip of the information system iceberg. When including all shadow applications in the picture, information systems today can already not guarantee the overall consistency, and rely upon humans to keep the whole together.

Collecting all shadow application data in a unified infrastructure may seem to aggravate inconsistency, but in reality it just reveals the present state, i.e. chaos. We think a unified infrastructure would provide additional leverage to the previously mentioned human factor in at least two ways.

In the consumer-space, "social" mechanisms like tagging, rating, voting, and targeted sharing have proven effective in organizing huge repositories of consumer-contributed data [12]. In a business environment, users could organize application elements through similar mechanisms. We think dealing with authenticated professionals is an even more beneficial setting than the consumer space for social technologies to apply, and envision *social information systems* where elements are contributed from the bottom up, shared with other Actors, ranked and improved through social feedback mechanisms and eventually gradually "promoted" to more central perspectives, providing a form of *evolutionary development* [13].

As opposed to today's situation where shadow applications are mostly disconnected from their parent applications and extremely heterogeneous in their implementation, a unified architecture would make the continuous evolution and the associated divergence *observable*. Indicators could be envisioned (number of extensions, number of unimports…) and dashboards built to monitor application evolution. Pattern-matching techniques could be used to automatically detect convergence opportunities [14, 15] and notify the owners of the candidate elements, fostering convergence discussions.

6.2 Impact on Collaboration

Although the goal of the proposed architecture is to make evolution a continuous process, introduction of significant chunks still requires traditional projects.

From a functional point of view, the painful and hazardous process of elaborating the union of divergent requirements could be replaced by the identification of the intersection, containing only the elements all stakeholders agree on, and then spawn smaller groups to discuss the next level of detail, thus reducing the risk of conflict and communication overhead. We think Perspectives would thus contain the various layers of *boundary objects* [16] around which actors collaborate.

From a software engineering point of view, private spaces could help in integrating running development projects with live production environments, facilitating continuous integration and delivery [16]. Boundaries between mockup, prototype, beta and production environments could be smoothened and concurrent development made easier, as well as quick experimentation encouraged.

6.3 Evaluation in the Real World

One of the challenges of this work is to find suitable ways to evaluate the underlying concepts of social information systems. A standard approach would be to deploy a robust implementation with a small group of users, and study its usage. However, if it were deployed in this fashion, it would become "just one more shadow application", and many of the benefits of a perspective-centric system would be lost. On the other hand, the approach is unfamiliar enough to both potential users and IT organizations that a major deployment would be difficult to accomplish. As illustrated by the aforementioned discomfort of the IT professionals, this requires a significant shift in thinking by IT and line-of-business managers about how crucial data is stored and managed. Introducing a perspective-centric system thus requires not only technological innovation, but also a high degree of organizational open-mindedness.

6.4 Challenges and Further Work

A real deployment of such a social architecture would almost certainly exhibit a high degree of coupling of its elements, making the system vulnerable to the evolution of central elements. However, since all dependencies are explicit, evolution policies could be defined. For example, if a high-level perspective deletes an element, it could be marked as orphan and be proposed to adoption to owners of perspectives which import or extend it.

We think a significant number of common business application features can be implemented in a generic way in the form of *functional aspects* [16] to be *applied* by an end-user while building his model. For example, if a particular attribute demands traceability this could be a single checkbox on the model's form, a simple boolean annotation on the model itself, and could tell a repository to produce history records with timestamp, user, and previous value. We are working on more complex aspects like lifecycle management and authorization.

The manipulation of model and instances through the same interface presents both the opportunity to leverage contributions from people without modeling skills and the risk to confuse them. Beyond the prototype's naïve forms for model manipulation, we consider usability for contributors with a broad spectrum of software skills a challenge. For programmers, the development of robust application code on top of a dynamic foundation is not trivial, and needs appropriate programming language bindings.

Other challenges are not new but rather inherited from the present situation. As an example, a user could define an extension concatenating two attributes, and export this extension to colleagues who do not have permission to see the initial data. This is similar to what happens when people extract confidential data in today's shadow applications, breaking the initial authorization mechanism. A perspective-centric system would actually improve on this situation; by having a complete view of all the attributes, a system would be able to detect and warn about possible permission violations.

At a higher level, perspective-centric architectures present a number of interesting challenges, like monitoring and convergence mechanisms, and adapting the consumer-space social recommendation mechanisms to application elements in a business environment.

7 Related Work

We consider the work presented in this paper a novel combination of existing approaches. Shadow applications are a widely known but widely accepted problem. They are frequently mentioned when studying information system agility [19] or dissatisfaction with business software [9], but not always considered as a problem [3].

Situational applications are enterprise applications built on-the-fly by business units to solve a specific business problem [10], and can be considered a superset of shadow applications. Situational applications have attracted recent interest from enterprise mashup researchers [20] who aim at allowing end users to integrate and combine services, data and other content [21] to bridge the IT/business gap. Mashups can be interpreted as an evolution of service-oriented architectures [22] which expose business functionality as standard and composable services.

Mashups are part of the broader topic of end-user development [23, 24], which advocates the empowerment of end-users to implement their own specific requirements, and has intensively studied spreadsheets [4, 25] and more recently collaborative and social aspects in enterprise settings [26].

Model-Driven Engineering (MDE) [27] elevates the level of abstraction at which software is developed, turning models into central and productive artifacts, with a specific models@runtime branch focusing on model interpretation. The Software Language Engineering (SLE) [28] and Domain-Specific Languages (DSL) [29] domains, related to MDE by the heavy reliance on meta-models, focus on domain expert involvement in software development and configuration through specific expert-friendly representations.

The Component-Based Software Engineering [30] community is actively researching robust dynamic systems, where components can appear and disappear during execution. It provides foundation concepts and technologies for making a social application cope with dynamic elements and services of variable reliability.

Linked Data [31] integrates distributed, loosely coupled and independently managed repositories of persistent entities, but targets an internet-wide database and mostly-read access.

Social Software Engineering focuses on the understanding of the human and social aspects of software engineering. It covers both the social aspects in the software engineering process and the engineering of social software [14].

In the Requirements Engineering domain, studies on ViewPoints [15] have focused on capturing divergent concerns but aim at reconciling these at the specification and design level. More recent work proposes to apply social mechanisms like voting and commenting [32], but restrict these to the requirements elicitation phase.

The tailoring of enterprise systems, from simple configuration to the modification of commercial code, is a topic of sufficient complexity for [11] to propose a typology.

Recent interest in cloud computing has yielded research in multi-tenancy [33], a way to configure the same software installation for various isolated organizations.

8 Conclusions

In this paper we have presented an alternative architecture for business applications designed to reduce shadow application proliferation. We have described the main characteristics of shadow applications, the causes of their emergence, and have proposed an architecture principle to defuse this phenomenon based on an isolation mechanism we call *perspectives*. We have presented our prototype, our first results on real-life use cases and the encouraging feedback it has received.

We have discussed our broader vision of a social information system leveraging the collective intelligence of an organization's employees, and the possibility of democratic evolution through the use of social mechanisms.

We have no silver bullet claim, rather a potentially interesting paradigm worth exploring. We have no revolution claim either, merely an original combination of existing approaches and a generalization of business application configuration mechanisms. This is enabled by continuously growing processing power versus fairly stable core requirements of business applications, a better understanding of distributed systems, and recent social technologies.

Acknowledgements. This work has been funded by Nano-2012 grant MoDeSI. The authors would like to thank the participating interviewees for their time and helpful comments.

References

1. Newell, S., Wagner, E.L., David, G.: Clumsy Information Systems: A Critical Review of Enterprise Systems, Agile Information Systems. Elsevier, Burlington (2007)
2. Zarnekow, R., Brenner, W., Pilgram, U.: Integrated Information Management: Applying Successful Industrial Concepts in IT (2006) ISBN 978-3540323068
3. Handel, M., Poltrock, S.: Working Around Official Applications: Experiences from a Large Engineering Project. In: CSCW (2010)
4. Nardi, B.A., Miller, J.R.: An ethnographic study of distributed problem solving in spreadsheet development. In: CSCW (1990)
5. Hordijk, W., Wieringa, R.: Rationality of Cross-System Data Duplication: A Case Study. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 68–82. Springer, Heidelberg (2010)
6. Gotts, I.: A New Cloud: The Stealth Cloud?. http://www.cio.com/article/630164 (2010)
7. Kling, R.: Cooperation coordination and control in computer-supported work. Commun. ACM **34**, 12 (1991)
8. Ackerman, M.S.: The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. Hum. Comput. Interact. **15**, 179–203 (2000)
9. Hoyer, V., Stanoevska-Slabeva, K.: The Changing Role of IT Departments in Enterprise Mashup Environments. In: Feuerlicht, G., Lamersdorf, W. (eds.) ICSOC 2008. LNCS, vol. 5472, pp. 148–154. Springer, Heidelberg (2009)
10. Markl, V., Altinel, M., Simmen, D. and Singh, A., Data Mashups for Situational Applications. In: MBSDI (2008)
11. Brehm, L., Heinzl, A., Markus, M.L.: Tailoring ERP systems: a spectrum of choices and their implications. In: HICSS (2001)
12. Surowiecki, J.: The Wisdom of Crowds (2004) ISBN 978-0385503860
13. Mørch, A.: Evolutionary application development, reframing humans in information systems development, pp. 151–171. Springer, London (2011)
14. Ahmadi, N., Jazayeri, M., Lelli, F., Nesic, S.: A survey of social software engineering. In: IEEE/ACM ASE – Workshops (2009)
15. Sabetzadeh, M., Finkelstein, A., Goedicke, M.: ViewPoints. In: Laplante, P. (ed.) Encyclopedia of Software Engineering. Taylor & Francis, New York (2010)
16. Star, S.L.: The structure of ill-structured solutions : boundary objects and heterogeneous problem solving, distributed artificial intelligence, vol. 2. Morgan Kaufmann, Los Altos (1990)
17. Fowler, M.: Continuous delivery : reliable software releases through build, test, and deployment automation. Addison Wesley, Boston (2010) ISBN: 978-0-321-60191-9

18. Filman, R.E., Elrad, T., Clarke, S., Akşit, M.: Aspect-oriented software development (2008) ISBN 0-321-21976-7
19. Desouza, K.C. (ed.) Agile Information Systems: Conceptualization, Construction and Management (2007) ISBN 978-0-7506-8235-0
20. Hoyer, V., Fischer, M.: Market overview of enterprise mashup tools. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 708–721. Springer, Heidelberg (2008)
21. Bitzer, S., Schumann, M.: Mashups: An approach to overcoming the business/it gap in service-oriented architectures. In: Nelson, M.L., Shaw, M.J., Strader, T.J. (eds.) Value Creation in e-Business Management. LNBIP, vol. 36, pp. 284–295. Springer, Heidelberg (2009)
22. Watt, S.: Mashups - The evolution of the SOA: Situational applications and the Mashup ecosystem. http://ibm.com/developerworks/webservices/library/ (2007)
23. Nestler, T.: Towards a Mashup-driven End-User Programming of SOA-based Applications. In: iiWAS (2008)
24. Sutcliffe, A.: Evaluating the costs and benefits of end-user development. SIGSOFT SEN **30**(4), 1–4 (2005)
25. Spahn, M., Wulf, V.: End-user development for individualized information management: analysis of problem domains and solution approaches. In: Filipe, J., Cordeiro, J. (eds.) ICEIS 2009. LNBIP, vol. 24, pp. 843–857. Springer, Heidelberg (2009)
26. Ahmadi, N., Jazayeri, M., Lelli, F., Repenning, A.: Towards the Web Of Applications: Incorporating End User Programming into the Web 2.0 Communities. In: SoSEA (2009)
27. Schmidt, D.C.: Model-Driven Engineering. IEEE Comput. **39**(2), 25–31 (2006)
28. Kleppe, A.: Software Language Engineering: Creating Domain-Specific Languages Using Metamodels (2008) ISBN 9780321553454
29. Fowler, M.: Domain-Specific Languages, ISBN 978-0-321-71294-3 (2010)
30. McIlroy, D.: Mass-Produced Software Components, Software Engineering, Report on a conference sponsored by the NATO Science Committee (1968)
31. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semant. Web Inf. Syst. **5**(3), 1–22 (2009)
32. Lohmann, S., Dietzold, S., Heim, P., Heino, N.: A web platform for social requirements engineering. In: Software Engineering Workshops (2009)
33. Jansen, S., Houben, G.-J., Brinkkemper, S.: Customization realization in multi-tenant web applications: case studies from the library sector. In: Benatallah, B., Casati, F., Kappel, G., Rossi, G. (eds.) ICWE 2010. LNCS, vol. 6189, pp. 445–459. Springer, Heidelberg (2010)

Finding Semi-Automatically a Greatest Common Model Thanks to Formal Concept Analysis

Bastien Amar[1(✉)], Abdoulkader Osman Guédi[1,2,3], André Miralles[1,3], Marianne Huchard[3], Thérèse Libourel[4], and Clémentine Nebut[3]

[1] Tetis/IRSTEA, Maison de la Télédétection, 500 rue J.-F. Breton, 34093 Montpellier Cedex 5, France

[2] Université de Djibouti, Avenue Georges Clemenceau, BP: 1904, Djibouti, Djibouti(REP)

[3] LIRMM, University of Montpellier 2 et CNRS, 161, rue Ada, 34392 Montpellier Cedex 5, France

[4] Espace Dev, Maison de la Télédétection, 500 rue J.-F. Breton, 34093 Montpellier Cedex 5, France

amar.bastien@free.fr, aogdj@hotmail.fr, andre.miralles@teledetection.fr, marianne.huchard@lirmm.fr, therese.libourel@lirmm.fr, clementine.nebut@lirmm.fr

Abstract. Data integration and knowledge capitalization combine data and information coming from different data sources designed by different experts having different purposes. In this paper, we propose to assist the underlying model merging activity. For close models made by experts of various specialities on the same system, we partially automate the identification of a Greatest Common Model (GCM) which is composed of the common concepts (core-concepts) of the different models. Our methodology is based on Formal Concept Analysis which is a method of data analysis based on lattice theory. A decision tree allows to semi-automatically classify concepts from the concept lattices and assist the GCM extraction. We apply our approach on the *EIS-Pesticide* project, an environmental information system which aims at centralizing knowledge and information produced by different research teams.

Keywords: Formal concept analysis · Greatest common model · Information system · Environmental information system · Model factorization · Core-concept · Measuring station · Pesticide

1 Introduction and Problematics

Elaborating data models is a recurrent activity in many projects in different domains, for various objectives: building dictionaries of the domain, designing databases, developing software for this domain, etc. Usually, such models of the domain are required by several teams, dealing with different facets of the domain, and potentially stemming from different scientific domains. For example, in the IRSTEA institute (in which three of the authors work), the study

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 72–91, 2013.
DOI: 10.1007/978-3-642-40654-6_5,

of pesticide impact on environment involves specialists from different scientific domains: hydrology, agronomy, chemistry, etc.

In this paper[1], we address the issue of assisting this gathering activity, in the context of domain data models designed with UML class diagrams through the automated detection of common domain-concepts (with two levels of confidence) possibly enriched with new domain-concepts automatically extracted from the previous ones. This approach is based on Formal Concept Analysis (FCA), which is an exact and robust data analysis method based on lattice theory. We use FCA to detect commonalities, redundancies and introduce new abstractions, both inside the models taken individually (intra-model factorization), and inside two distinct data models taken jointly (inter-model factorization). The approach defined in this paper deals with two models, but more generally, it is able to identify the common domain-concepts of several models in order to help the designer to centralize these common concepts into a unique consolidated model (the GCM). This approach is under evaluation on a large project from the IRSTEA institute called Environmental Information System for Pesticides (*EIS-Pesticide*), in which two teams cooperate to build a domain data model. The Transfer team is specialized in the study of the pesticide transfer to the rivers, and the Practice team mainly works on the agricultural practices of farmers.

The rest of the paper is structured as follows. In Sect. 2 we introduce example models taken from the *EIS-Pesticide* project. In Sect. 3, we draw the main lines of our approach, and in Sect. 4, we provide a short introduction to Formal Concept Analysis (FCA). In Sect. 5 we explain how FCA is used on input models and how the resulting lattices are analyzed so as to provide the final user clear recommendations to build the greatest common model. In Sect. 6, we present schematically the process we have implemented into a case tool to generate automatically the greatest common model. In Sect. 7, we present the greatest common model corresponding to our models and then the application of our approach on larger models to evaluate its scalability. Section 8 presents the related work and Sect. 9 concludes the paper.

2 Running Example: The Two Models of Measuring Station

The Environmental Information System for Pesticides is a project [1,2] that has the objective to set up an information system allowing to centralize knowledge and information produced by Transfer and Practice teams (see Sect. 1). We illustrate our approach on a small subsystem representing part of the *measuring activity* on the catchment area (drainage basin): measuring stations by monitoring the major parameters involved in the transfer of the pesticides to the rivers.

[1] This article is an extension of *Using Formal Concept Analysis to Extract a Greatest Common Model* published in the ICEIS 2012 conference.

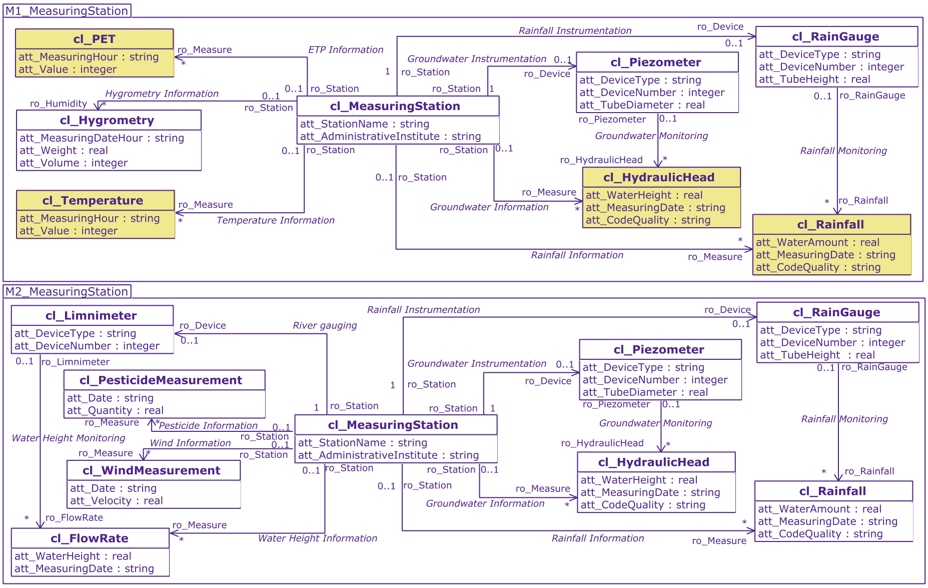


Fig. 1. The two data models of measuring station produced by the two teams.

Figure 1 shows the two data models of the measuring stations used in this study. They are produced by the two teams involved in the project. As these two models are very close, we have organized them by grouping at the r.h.s of measuring station (*cl_MeasuringStation*), the identical domain-concepts (that also have the same type of relationships). In this part of the model, the measured data are associated to the corresponding measuring device: the rainfall (*cl_Rainfall*) and the hydraulic head (*cl_HydraulicHead*) of the groundwater table are continuously recorded respectively by the rain gauge (*cl_RainGauge*) and by the piezometer (*cl_Piezometer*). Each of these measures is dated (see property *att_MeasuringDate*).

On the l.h.s. of *cl_MeasuringStation*, the model *M1_MeasuringStation* allows to record the data measured by a weather station of Météo-France (a french meteorological institute): temperature (*cl_Temperature*), hygrometry (*cl_Hygrometry*) and potential evapo-transpiration (*cl_PET*) of the short green crops.

These last domain-concepts are not in the model *M2_MeasuringStation* which has on the other hand a limnimeter (*cl_Limnimeter*) to measure continuously the flow rate (*cl_FlowRate*) of rivers. A technician is in charge to take samples in order to determine in laboratory the amount of pesticides in the water (*cl_PesticideMeasurement*). Finally, the wind velocity (*cl_WindMeasurement*) is a parameter coming from a weather station of Météo-France.

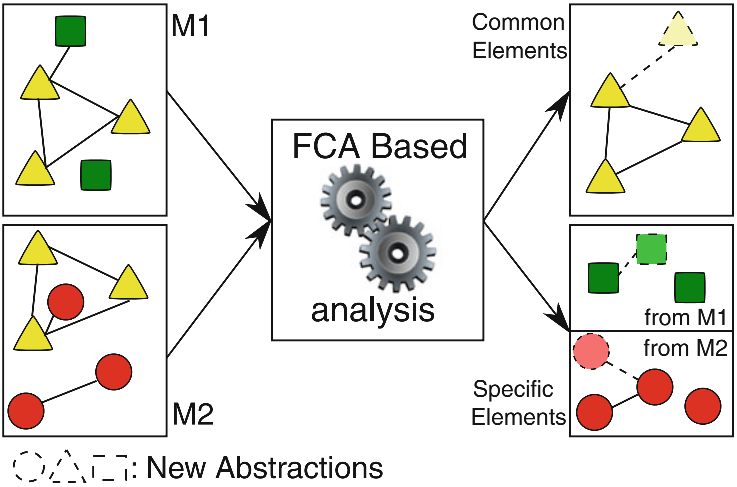


Fig. 2. A schematic overview of our approach (applied on one formal context).

3 Overview of the Proposed Approach

The main objective of our approach is to assist the task of gathering two or more models independently defined and thus potentially involving common concepts. For that we extract from initial models their Greatest Common Model (GCM).

The term "greatest common model" is chosen by analogy to the "greatest common divisor (GCD)" in arithmetic; it is more precisely defined in the following. Roughly, it contains all the common domain-concepts that are introduced in all the studied models, in a normal[2] (factorized) form.

The proposed approach is illustrated in Fig. 2. The input is two (or more) models for a domain, named M_1 and M_2. In a first time, the classes of the input models are described by their owned characteristics. Formal Concept Analysis (FCA) allows entities sharing characteristics to be grouped into formal-concepts, and results in lattices providing a hierarchical view of those formal-concepts. FCA can deal with several class characteristics: attributes, operations, roles... We apply FCA on several class descriptions, resulting in several lattices. These lattices allow the identification of common concepts, specific concepts and possibly new abstractions extracted from intra- or inter- model factorization. For instance, if we describe classes by their owned attributes, the resulting lattice (cf. Fig. 5) extracts the r.h.s. common domain concepts of Fig. 1. It also extracts new abstractions. Some new abstractions are present both in M_1 and M_2 (e.g. a *device* concept factorizes commonalities of rain gauge, and piezometer: inter-model factorization). Some other extracted abstractions are present only in a same model (e.g. a *dated measurement* concept factorizes pesticide and

[2] Here, we refer to the relational normal form used in database schema normalization, which has the same objective: eliminate redundancies.

wind measurements in M_2: intra-model factorization). For each lattice, we have two levels of confidence for those domain-concepts: domain-concepts which are very likely to be in the GCM, and others that have to be precisely analyzed, validated and named by the final expert. As we generate several lattices, the expert in charge of integration needs to follow a strategy for analyzing them. We propose to order the obtained lattices following the semantic hierarchy of the different factorization criteria. The lattices are then analyzed, so as to categorize formal-concepts and interpret them, if applicable, to form domain-concepts.

The domain-concepts recognized by the experts as being in the GCM are called the *core domain-concepts*. In Fig. 1, the domain-concepts to the right of *cl_MeasuringStation* are certainly core domain-concepts. The *greatest common model* (GCM) is defined as the largest model factorizing the core domain-concepts of several models.

4 A Short Introduction to Formal Concept Analysis

Formal Concept Analysis (FCA) [3] is a theoretical framework for conceptual classification, data analysis and clustering based on lattice theory [4]. FCA is applied in many domains including knowledge structuring, information retrieval, data mining, class model refactoring, or software analysis. FCA studies entities described by their characteristics to discover formal-concepts which are maximal groups of entities sharing maximal groups of characteristics. A partial order based on the entity set inclusion provides a lattice structure (the concept lattice) and organizes concepts by specialization.

A *formal context* K is a triple[3] $K = (E, C, R)$, where E is the set of entities and C the set of characteristics that describe these entities. Relation $R \subseteq E \times C$ associates an entity with its characteristics: $(e, c) \in R$ when entity e owns characteristic c. For example, Table 1 shows the formal context of the submodel highlighted in Fig. 1 (limited to the four classes *cl_PET, cl_Temperature, cl_HydraulicHead* and *cl_Rainfall*). The characteristics of the classes (the entities) are the names of their owned attributes.

A *formal-concept* is a pair $(Extent, Intent)$ where $Extent = \{e \in E | \forall c \in Intent, (e, c) \in R\}$ and $Intent = \{c \in C | \forall e \in Extent, (e, c) \in R\}$. These two sets represent the entities that own all the characteristics (extent) and the characteristics shared by all entities (intent). The specialization order between two formal concepts is given by the following equivalence: $(Extent\_1, Intent\_1) < (Extent\_2, Intent\_2) \Leftrightarrow Extent\_1 \subset Extent\_2$ (equivalently $Intent\_2 \subset Intent\_1$).

In a lattice, there is an ascending inheritance of entities and a descending inheritance of characteristics. The simplified intent of a formal concept is its intent without the characteristics inherited from its super-concept intents. The simplified extent is defined in a similar way.

Nota: in this article, we distinguish simplified extent from extent. When it is not specified, we are talking about (complete) extent.

[3] In the literature, standard notation is $K = (G, M, I)$. We use $K = (E, C, R)$ for readability reasons and to get a better understanding toward our thematic partners.

Table 1. The formal context of the reduced model.

	att_ MeasuringHour	att_ Value	att_ Water Amount	att_ MeasuringDate	att_ Code-Quality	att_ Water-Height
cl_ PET	×	×				
cl_ Temperature	×	×				
cl_ Rainfall			×	×	×	
cl_ HydraulicHead				×	×	×

For readability reasons, all lattices presented in this paper show simplified extents and intents.

Figure 3 shows the concept lattice built from the formal context presented Table 1. Each formal-concept is represented by a box in three parts: the first part contains the generated name of the formal-concept, the second part contains its simplified intent, and the last one contains its simplified extent. Let us consider `Concept_17`: it represents entities (classes) described by the characteristic *att_WaterHeight* and by the characteristics inherited from its super-concepts: *att_MeasuringDate* and *att_CodeQuality* (from `Concept_16`). The extent and the simplified extent of `Concept_17` is the set containing class *cl_HydraulicHead*. Let us consider `Concept_16`: its extent is composed of the set containing *cl_HydraulicHead* and *cl_Rainfall* (inherited from `Concept_15` and `Concept_17` respectively).

In this work, we are interested in three categories of formal-concepts that form a partition of the set of formal-concepts:

Definition 1 (Merged Formal Concepts). Merged formal concepts *have more than one entity in their simplified extent. This means that all entities in the extent are described by exactly the same set of characteristics.*

In Fig. 3, `Concept_13` is a merged formal concept: the two classes of its simplified extent, namely *cl_PET* and *cl_Temperature*, are (exactly) described by both characteristics *att_MeasuringHour* and *att_Value.*

Definition 2 (New Formal Concepts). New formal concepts *have an empty simplified extent. These are new, more abstract, concepts, factoring out characteristics common to several formal-concepts.*

In Fig. 3, `Concept_16` is a new formal concept, factoring out characteristics of both `Concept_15` and `Concept_17`.

Definition 3 (Perennial Formal Concepts). Perennial formal concepts *have one and only one entity in their simplified extent.*

In Fig. 3, both `Concept_15` and `Concept_17` are perennial. In this article, merged, new and perennial formal concepts are respectively annotated, in the figures, **M**, **N** and **P** at the right-top corner.

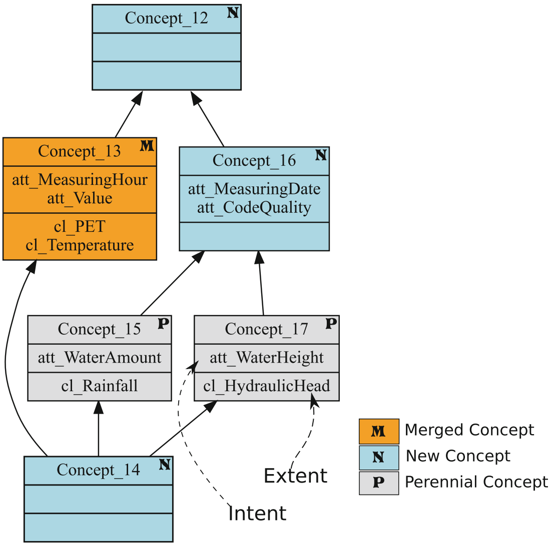


Fig. 3. Class/attribute name lattice: result of FCA on Table 1.

5 Applying Formal Concept Analysis to Extract Candidates for the Greatest Common Model

In this section, we propose a methodology based on two automatic steps that uses Formal Concept Analysis (FCA) and an interactive process to extract the greatest common model of two input models. Given two models M_1 and M_2:

- From several characteristics (owned attributes, roles...), we compute the lattices resulting from FCA applied to several formal contexts extracted from the disjoint union of the two input models $M = M_1 \oplus M_2$. The different formal contexts are obtained by considering classes described by different characteristics (class names, attributes, roles, etc.).
- The concepts of these lattices are analyzed thanks to a decision tree based on the analysis of the concept extent, and we obtain lists of concepts.

In the interactive process, these lists are exploited to assist the expert to build the greatest common model. The next subsections precisely describe two automatic steps.

5.1 Applying FCA on the Two Models

As explained in Sect. 4, formal contexts describe entities by characteristics. Many different formal contexts can be extracted from a class model: it has to be defined which model elements are chosen to be the studied entities, and which features of those model elements are chosen to be their studied characteristics. Here we

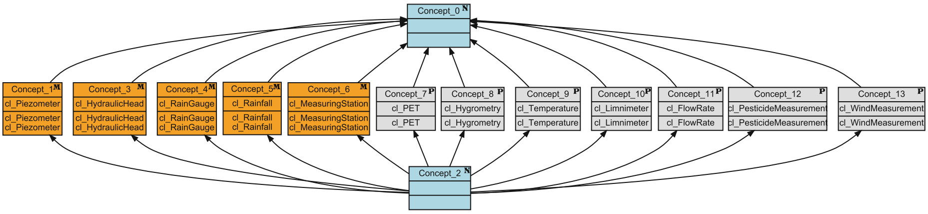


Fig. 4. The class/class name concept lattice.

focus on three formal contexts extracted from the disjoint union of input models $M = M_1 \oplus M_2$:

1. the formal context of classes described by their name,
2. the formal context of classes described by their attributes,
3. the formal context of classes described by their attributes and by their roles.

Figure 4 presents the lattice obtained with the formal context of classes described by their name (*class/class name* lattice). This lattice groups in a concept the set of classes sharing the same name. For example, the merged concept `Concept_1` represents the set of classes (in extent) sharing the name (in intent) *cl_Piezometer*. In other words, FCA merged in a single concept classes that have a same name. Classes that are not duplicated in the models M_1 and M_2 remain in a perennial concept, like the *cl_PET* class in `Concept_7`. In inter-model factorization, the three categories of concepts described in Sect. 4 exist: the merged concept `Concept_1` has more than one entity in its simplified extent. In a similar way, the perennial concept `Concept_7` (*cl_PET*) has exactly one element in its extent. Later we will see the case where new formal concepts appear.

Figure 5 presents the lattice obtained with the formal context of classes described by the names of their owned attributes (*class/attribute name* lattice). In this lattice, a formal concept thus is a group of classes (extent) sharing a group of attribute names (intent). The lattice contains new formal concepts ($simplified\ extent = \emptyset$), e.g. `Concept_47`, that represents a new abstraction: things that are dated.

Figure 6 presents the lattice obtained with the formal context of classes described by the names of their owned attributes and roles (*class/attribute-role name* lattice). UML associations are taken into account in this lattice through those roles. For example, class *cl_FlowRate* has attribute *att_WaterHeight* and role *ro_Station* in association *Water Height Information*. The new formal concept `Concept_30` represents the classes that are linked with a Station *via* the role *ro_Station*. Class *cl_FlowRate* belongs to the extent of this concept.

5.2 Analysis of the Lattices

In this section, we present the analysis of the lattices using a decision tree to classify each concept. First, the *class/class name* lattice must be analyzed. This

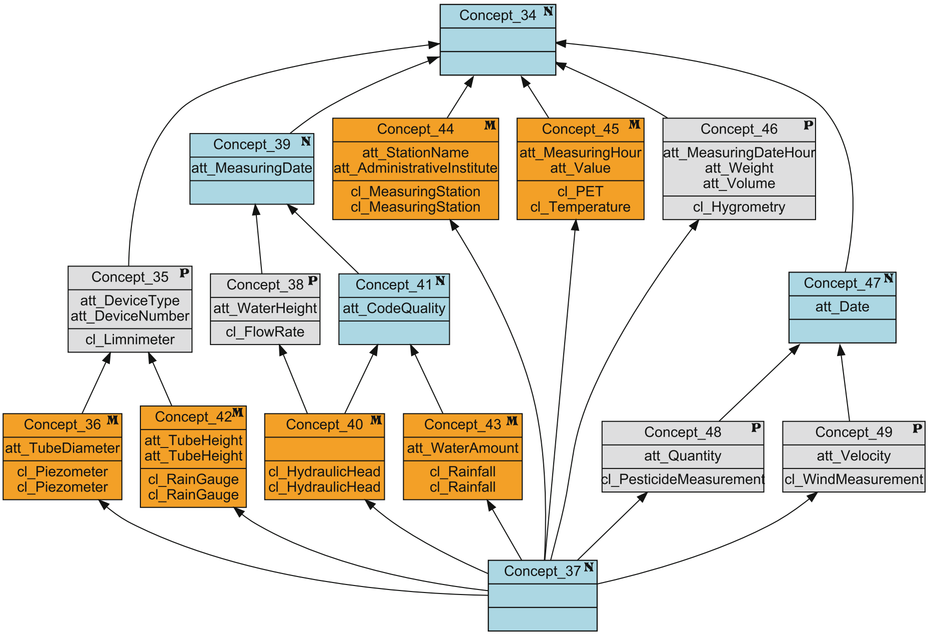


Fig. 5. The class/attribute name concept lattice.

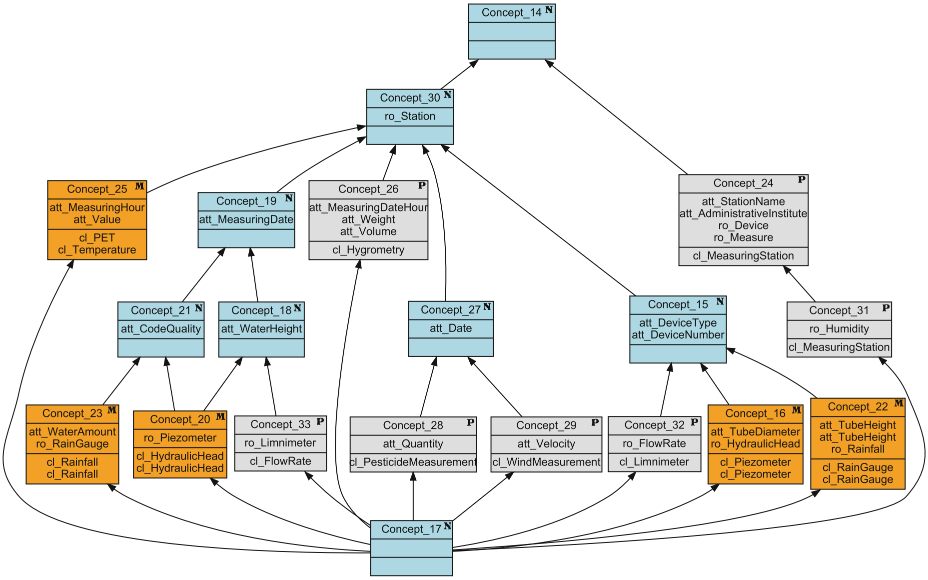


Fig. 6. The class/attribute-role name concept lattice.

lattice allows the designer to group classes that have a same name. Then, we analyze the *class/attribute name* lattice that allows us to find attribute-based factorizations. As we will see, the *class/attribute-role name* lattice can be a considerable help to refine the decisions about factorization.

For each formal concept $Co_k = (E_k, I_k)$, the concept type (merged, new or perennial) and the *complete* extent E_k have to be analyzed and the concept has to be included in one of these lists:

- L_{GCM} is the list of core-concepts that will be included in the greatest common model.
- L_{pGCM} is the list of potential (candidate) core-concepts to be validated by an expert to be in the greatest common model.
- L_{M_1} and respectively L_{M_2} are the lists of domain concepts specific to M_1 (resp. M_2).
- L_{nM_1} and respectively L_{nM_2} are new domain concepts specific to M_1 (resp. M_2), factorizing existing domain concepts. These domain concepts are not intended to be in the greatest common model, but they can be presented to experts to improve the factorization of M_1 (resp. M_2).

Figure 7 presents the decision tree: we define $\mathcal{C}_{M_i}$ (resp. $\mathcal{C}_{M_j}$) as the set of classes in the model M_i (resp. M_j), and the decision tree is designed for two models M_i and M_j where $i \neq j$. As we apply FCA with classes as entities (characteristics being class name, attributes, and/or roles), the extent of a concept contains only classes. For each concept, we first check if the concept is a *merged concept*, a *new concept* or a *perennial concept* (nodes 1, 8 and 12 respectively in the decision tree of Fig. 7) as defined in Sect. 4.

Analysis of Merged Concepts

If the concept is a merged concept, then three cases are possible: its extent

1 → $Merged\ Concept$
  2 → $E_k \cap \mathcal{C}_{M_i} \neq \emptyset \wedge E_k \cap \mathcal{C}_{M_j} \neq \emptyset$
    3 → $|E_k \cap \mathcal{C}_{M_i}| = 1 \wedge |E_k \cap \mathcal{C}_{M_j}| = 1 \rightarrow L_{GCM}$
    4 → $|E_k \cap \mathcal{C}_{M_i}| = 1 \wedge |E_k \cap \mathcal{C}_{M_j}| > 1 \rightarrow L_{pGCM}$
    5 → $|E_k \cap \mathcal{C}_{M_i}| > 1 \wedge |E_k \cap \mathcal{C}_{M_j}| > 1 \rightarrow L_{pGCM}$
  6 → $E_k \cap \mathcal{C}_{M_i} \neq \emptyset \wedge E_k \cap \mathcal{C}_{M_j} = \emptyset \rightarrow L_{nM_i}$
  7 → $E_k \cap \mathcal{C}_{M_i} = \emptyset \wedge E_k \cap \mathcal{C}_{M_j} = \emptyset \rightarrow Inconsistent$
8 → $New\ Concept$
  9 → $E_k \cap \mathcal{C}_{M_i} \neq \emptyset \wedge E_k \cap \mathcal{C}_{M_j} \neq \emptyset \rightarrow L_{pGCM}$
  10 → $E_k \cap \mathcal{C}_{M_i} \neq \emptyset \wedge E_k \cap \mathcal{C}_{M_j} = \emptyset \rightarrow L_{nM_i}$
  11 → $E_k \cap \mathcal{C}_{M_i} = \emptyset \wedge E_k \cap \mathcal{C}_{M_j} = \emptyset \rightarrow Bottom$
12 → $Perennial\ Concept$
  13 → $E_k \cap \mathcal{C}_{M_i} \neq \emptyset \wedge E_k \cap \mathcal{C}_{M_j} \neq \emptyset \rightarrow L_{pGCM}$
  14 → $E_k \cap \mathcal{C}_{M_i} \neq \emptyset \wedge E_k \cap \mathcal{C}_{M_j} = \emptyset \rightarrow L_{M_i}$
  15 → $E_k \cap \mathcal{C}_{M_i} = \emptyset \wedge E_k \cap \mathcal{C}_{M_j} = \emptyset \rightarrow Inconsistent$

Fig. 7. Decision tree.

contains elements from both models M_i and M_j (node 2), its extent contains only elements from M_i (node 6), or its extent is empty (node 7).

If the concept extent contains elements from both models, the cardinality of the intersection between the extent and the set of model classes has to be checked. In the first case, the extent contains only one class from M_i and only one class from M_j (node 3) like `Concept_1` in the *class/class name* lattice, Fig. 4. Then a corresponding domain concept should be added in L_{GCM}: it can be considered as a core-concept – a domain concept common to both models. If the extent contains only one class from M_i and several classes from M_j (node 4), or several elements from both models (node 5), then it should be put in the L_{pGCM} list: it is a potential core-concept, but an expert intervention is necessary. He or she can choose to merge or factorize duplicated classes if they are semantically closed, in a same model (intra-model factorization), and relaunch the process to extract the greatest common model. He or she can also consider these classes as specific domain concepts and keep them in the specific model.

If the merged concept contains only classes from M_i (node 6), like the `Concept_45` in the Fig. 5, it should be added to the L_{nM_i} list. Figure 8 presents an excerpt of the class attribute name lattice. We have added some classes next to their corresponding concepts, and we have highlighted the factorized attributes to illustrate the explanation.

Its extent contains a group of elements coming from a same model and that are described exactly by the same characteristics. It can be presented to an expert to improve the model M_i, but it is not a core-concept (they are in one model only). In the case of `Concept_45`, FCA suggests to merge the classes *cl_PET* (representing the Potential Evapo-Transpiration) and *cl_Temperature*. In this special case, these two classes are semantically different, and the expert do not want to factorize them, but in other situations he could consider this factorization to be interesting.

The node 7 describes concepts wherein the extent does not contain classes from M_i and M_j. This is inconsistent: by definition, a merged concept extent contains at least two elements (*cf* Definition 1).

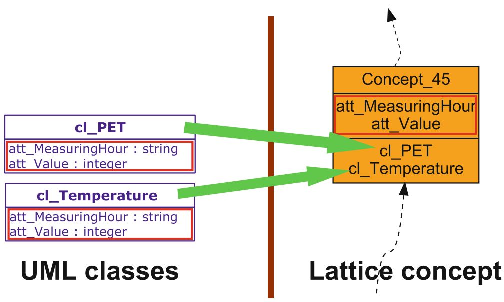


Fig. 8. Excerpt of the class/attribute name concept lattice and corresponding classes.

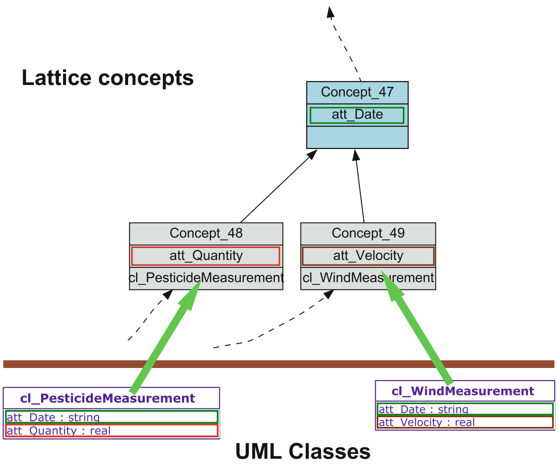


Fig. 9. Excerpt of the class/attribute name concept lattice and corresponding classes.

Analysis of New Concepts
If the concept is a new concept (*cf.* Definition 2, node 8), and if its extent contains elements from both models M_i and M_j (node 9) then the concept has to be put in the L_{pGCM} list: it is a potential factorization of concepts defined in M_i and M_j, so it is potentially a core-concept. Experts have to decide if this factorization is valid and if this new concept has to be included in the greatest common model. `Concept_39` in Fig. 5 is an example of this type of concept. In our case study, the expert validates this concept to be a greatest common model concept.

If the new concept extent contains only classes from one model, it can be added in the L_{nM_i} list (node 10 in the decision tree). This concept corresponds to an intra-model factorization. A concrete case of intra-factorization is presented in Fig. 9: the `Concept_47` represents things that are dated in M_2 by factorizing the `date` attribute. This kind of concept is not a core-concept and should not be included in the greatest common model. It can be presented to the M_i designer in order to raise the quality of its model by a new factorization.

If the new concept extent (node 11 in the decision tree) does not contain elements from M_1 nor M_2, this means that this is the concept *Bottom*. Concept *Bottom* is present in each lattice (concepts `Concept_2`, `Concept_37` and `Concept_17`). It represents elements that own all attributes and should not be used in our re-engineering process. Instead, the top concept cannot be inferred only by extent analysis and it may appear in each branch of the tree. Depending on the configuration of the analyzed models, this concept may be relevant and it is classified as other concepts.

Analysis of Perennial Concepts
Node 13 in the decision tree describes perennial concepts that have in their extent classes from M_i and M_j, like `Concept_35` in Fig. 5. This means that there

is a potential factorization of `Concept_36` and `Concept_42`, and this factorization already exists, *cl_Limnimeter* in our example. This kind of concept has to be presented to the expert, it is thus added to the L_{pGCM} list. In our example, the designer can make *cl_Limnimeter* be a super-class of *cl_Piezometer* and *cl_Raingauge*, but this decision is not semantically valid: a piezometer *is not* a limnimeter. An analysis of the lattice of classes described by their attributes and role names (cf. Fig. 6) shows that it is better to create a new superclass (`Concept_15`) of data instrumentation, factorizing the three classes *cl_Limnimeter*, *cl_Piezometer* and *cl_RainGauge*. In this case, the lattice of classes described by their attributes/roles names is useful to help the designer to take a decision.

If the perennial concept extent contains only classes from M_i (node 14) then it is a M_i domain specific concept. This concept must be added to L_{M_i}. For example, concepts `Concept_7`, `Concept_8`, `Concept_48`, and `Concept_49` are domain concepts specific to M_i. `Concept_48`, and `Concept_49` are presented in Fig. 9 with their corresponding classes *PesticideMeasurement* and *WindMeasurement*. These classes are not factorized and remain specific concepts of M_2.

A perennial concept cannot have an empty extent (node 15): the Definition 3 specifies that a perennial concept has one (and only one) element in its extent.

From both L_{GCM} and L_{pGCM} lists, the expert has to select the core-concepts that will be included in the GCM.

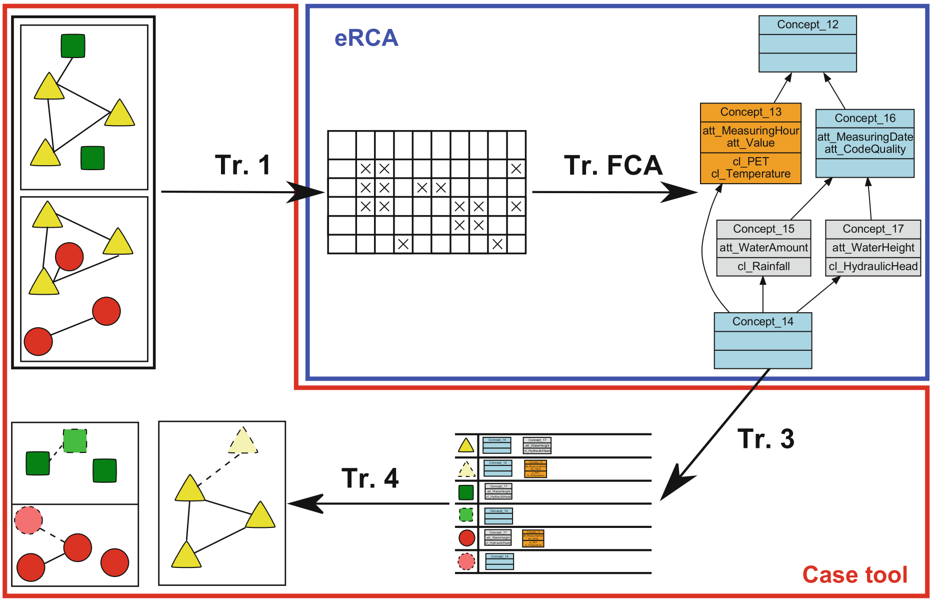


Fig. 10. Detailed implementation of the working process of our approach.

6 Working Process of Our Approach and Its Implementation

The working process of our approach is structured around four major transformations (cf. Fig. 10). The first one (Tr. 1) converts UML models into the different types of formal contexts which are the inputs of FCA. The second one (Tr. FCA) is the FCA process which transforms the formal contexts into lattices. The next transformation (Tr. 3) generates the various lists of domain-concepts in accordance with the decision tree. Finally, the last transformation (Tr. 4), called "GCM build" is under development. It will use the classification of concepts we introduced and the decisions taken by the developer to build the UML model. For the *Measuring Station* models (cf. Fig. 1), we have manually extracted the greatest common model from the classification output because these models are small. The manual extraction is impossible from the *EIS-Pesticide* model because it contains more than 200 entities.

This working process is implemented as a profile into the Objecteering case tool[4] which encapsulates the eRCA[5] component. Figure 10 presents the association of these two components.

Objecteering is a UML modeler edited by Objecteering Software and integrating model driven engineering principles. Its main functionalities are: creation and edition of UML models, code generation and reverse engineering of existing projects. Objecteering allows designers to extend the proposed functionalities by creating plug-ins called Model Driven Approach Component (MDAC). These plug-ins can manipulate models thanks to an API reifying the UML meta-model, and tools are available to enhance the original user interface.

eRCA is a framework that ease the use of formal and relational concept analysis. Originally designed as an Eclipse plug-in, it is also available as a standalone version through a *jar* library, so as to be used in classical Java projects.

In practice, the *MeasuringStation* and *EIS-Pesticide* models have been designed in Objecteering. Thanks to the MDAC extension system, we navigate through these models and generate formal contexts. The FCA transformation applies formal concept analysis to the formal contexts and produces lattices. The eRCA component includes an API to manipulate these lattices, it is used in the MDAC of Objecteering to navigate through the lattices, using the decision tree to classify each concept.

7 Results

Figure 11 shows the model obtained by applying our approach: the final greatest common model of the M1 and M2 models (cf. Fig. 1). This GCM reflects also the

[4] http://www.objecteering.com/. The development has been done with Objecteering but we are migrating it to Modelio, the last version of Objecteering.

[5] http://code.google.com/p/erca/

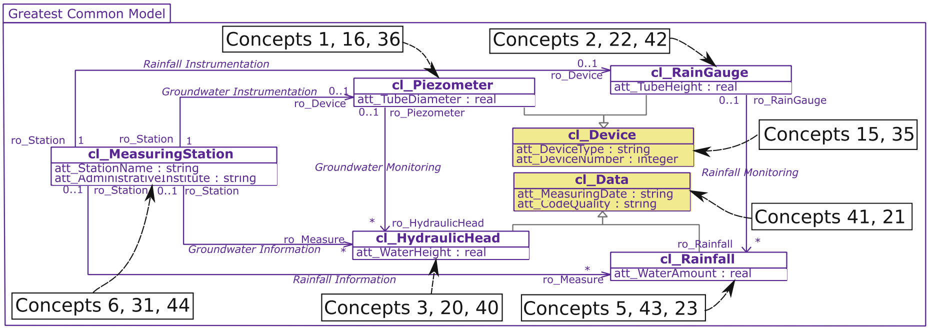


Fig. 11. The greatest common model of M1 and M2 models (cf. Fig. 1).

Table 2. Result of our approach on the *MeasuringStation* model.

	L_{GCM}	L_{pGCM}	L_{nM1}	L_{nM2}	L_{M1}	L_{M2}
class/class name	5	1	0	0	3	4
class/attribute name	5	5	1	1	1	2
class/attribute-role name	4	7	1	1	2	4

interpretation and the validation by an expert of the new concepts. We annotated classes by associated formal concepts that represent them in the lattices (cf. Figs. 4, 5, 6).

As expected, the same domain-concepts in both models M1 and M2 are present in the *GCM*: *cl_MeasuringStation*, *cl_Piezometer*, *cl_HydraulicHead*, *cl_RainGauge* and *cl_Rainfall*. They constitute the core-concepts of the *GCM* of M1 and M2. So, they are automatically added in the L_{GCM} list.

Our approach proposes a list of possible factorizations of domain-concepts in the L_{pGCM} list. The expert must validate the relevance of these concepts. In this example, two new concepts have been considered relevant. They are colored in Fig. 11.

The first corresponds to formal concepts `Concept_15` (cf. Fig. 6) and `Concept_35` (cf. Fig. 5) in the lattices. They factorize attributes *att_DeviceType* and *att_DeviceNumber*. This concept has been validated by experts as a new *cl_Device* class.

The second new concept corresponds to formal concepts `Concept_41` and `Concept_21` in the lattices. It factorizes both *att_MeasuringDate* and *att_CodeQuality* attributes. Similarly to the first new concept, experts validate this concept as a new *cl_Data* class.

Table 2 quantifies for each formal context the number of concepts in each list defined in the decision tree.[6]

[6] In these tables, new and merged concepts must be still validated by an expert.

Table 3. Result of our approach on the complete *EIS-Pesticide* model.

	L_{GCM}	L_{pGCM}	L_{nM1}	L_{nM2}	L_{M1}	L_{M2}
class/class name	111	34	0	0	1	1
class/attribute name	43	39	0	0	1	2
class/attribute-role name	68	119	0	0	8	9

In order to validate the scalability of our approach, tests have been done on two versions of the complete model from the *EIS-Pesticide* project (about 125 classes). Table 3 gives the number of concepts by list of the decision tree (see footnote 6).

With the class/class name and class/attribute name lattices, experts have to analyze and to validate between 34 and 39 concepts present in the L_{pGCM} list. They can obtain more precision (with also more analysis work) with the class/attribute-role name lattice, where 119 potential GCM concepts are proposed. We are currently working to assist the expert in this analysis task [5]. We can also deduce from these results that the two versions of pesticide model are very close: there are only few specific concepts.

8 Related Work

FCA is used to improve the abstraction quality and the duplication elimination in class models in various domains (software engineering, ontology mapping or merging). This feature led us to propose the construction of a GCM to capitalize the knowledge of various domains.

Many variants have been studied, which take into account different characteristics for classes (the entities or domain-concepts in this framework): attribute names, attribute types, operation names, operation signatures, type specialization... The relevance of this approach is related to the properties satisfied by the class model after refactoring: all duplications are eliminated and the specialization relation between formal concepts meets the inclusion of features in the class model. These previous approaches only focus on intra-model factorization. In this paper, we use FCA for *inter-model* factorization, and we need to analyze differently the lattices, to identify categories of formal-concepts useful to build the greatest common model of several input class models. We define a guide for the expert to assist the building of the GCM. Indeed, in this work, we assume that if two characteristics have the same name, then these two characteristics are identical. Some work includes semantic analysis [6,7].

In software engineering, FCA has been used to build and maintain class hierarchies [8–10]. In this paper, our objective is different, we want to find common and specific parts between several models. The management of similarities and differences between models has been studied in the domain of model versioning [11]. The Smover tool uses direct comparison between a model and its previous version to detect syntactic and semantic conflict [12]. In order to manage model conflicts in a distributed development context, the work presented in [13] proposes the use of a difference model to store differences between two versions of a same model [14]. These methods allow to show differences between models, but they don't aim to propose automatic core-concept detection. In the approach described in [15], models and diagrams are considered as syntax trees, which allows the authors to design a difference operation between models. Compared to the domain of model versioning, we aim to present the GCM in a normal (factorized) form. This is why FCA is more suitable for our problem.

Formal concept analysis has been used to perform ontology mapping or merging, which is an issue close to ours [16,17]. The approach proposed by [18] uses FCA and linguistic analysis to merge ontologies in a semantic web context. In order to align ontologies, there are approaches that use a similarity measure, based on FCA [19] or on ontologies internal structure and association rule mining [20]. All these works aim to perform ontology mapping, while we work to extract the mapping result and to abstract new domain-concepts.

Since the early 80s, the database domain has studied the problem of schema integration and data matching, particularly in the database integration context. The aim of database integration context is to produce the global schema of a collection of databases [21–23]. Producing such a global database schema is an issue close to the extraction of a greatest common model in the sense that the search for identical concepts in different schemas is a necessary step. There are a lot of work dealing with this problematic in the literature. Generally, integration is composed of different steps: schema transformation, correspondence investigation and schema integration. Our work focuses on correspondence investigation and schema integration [24]. The integrated schema includes the GCM and the specific part of the initial schemas. There are two groups of solutions to semi-automatically find matches: rule-based solutions and learning-based solutions. Our approach is similar to rule-based solutions: we search similarity between several model elements based on their characteristics [25]. Unlike these approaches, the use of FCA allows to choose with finesse the way to describe the characteristics that we consider. In this article, we focus on the description of classes by their name, attribute name or role name, but FCA opens many other possibilities.

9 Conclusions

During domain modeling activity, several teams with different scientific skills usually make different models of a same domain. Each specialized team models the part of the domain model it is familiar with, and finally, a unique, consolidated domain model has to be built. This model integration requires the identification of the common domain-concepts that are present in the various specialized models.

Our contribution in this paper is an approach to assist the gathering task for several given class diagrams describing the domain. The proposed methodology is based on *Formal Concept Analysis* and the analysis of the formal-concepts using a decision tree. It allows the production of a Greatest Common Model in a normal (factorized) form. Our approach proposes two levels of confidence for candidate GCM concepts: domain-concepts which certainly will be in the GCM, and domain-concepts that have to be precisely analyzed, validated and named by experts. Moreover, the approach identifies specific-concepts and proposes possible new concepts that factorize the original models. We have validated the scalability of our approach by applying it on two versions of the *EIS-Pesticide* model, versions containing about 125 classes. The results of our approach were analyzed, validated and used by A. Miralles, co-author of this paper, who has a dual expertise: computer science and spraying application techniques of pesticides [2,26,27]. The contribution presented in this paper is an important step toward knowledge capitalization, which we applied here to the agriculture domain, for the IES-Pesticide project.

One of the major perspectives to our work is to improve the GCM through the use of Relational Concept Analysis (RCA), which is an FCA extension that will allow us to work more precisely on the relationships (UML associations) between domain-concepts. In our running example, the use of RCA would enable factorizing the *Rainfall Instrumentation* and the *Groundwater Instrumentation* associations with a new association connecting the new domain-concept *cl_Device* with the *cl_MeasuringStation* class. Similarly, RCA would extract a new association between the new *cl_Data* class and *cl_MeasuringStation*, factorizing both *RainFall Information* and *GroundWater Information* associations.

Another perspective is the use of natural language processing techniques to improve the name-based description of elements (classes, attributes, roles, etc). The knowledge of semantic relations like hyperonymy, synonymy, or homonymy between terms will refine the analysis of domain-concepts.

References

1. Pinet, F., Miralles, A., Bimonte, S., Vernier, F., Carluer, N., Gouy, V., Bernard, S.: The use of uml to design agricultural data warehouses. In: International Conference on Agricultural Engineering (AgEng 2010), pp. 1–10 (2010)
2. Miralles, A., Pinet, F., Carluer, N., Vernier, F., Bimonte, S., Lauvernet, C., Gouy, V.: EIS-Pesticide: an information system for data and knowledge capitalization and analysis. In: Euraqua-PEER Scientific Conference 1, 26 Oct 2011–28 Oct 2011, Montpellier, FRA (2011)
3. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundation. Springer, Berlin (1999)
4. Birkhoff, G.: Lattice theory. American Mathematical Society, Providence (1940)
5. Osman Guedi, A., Miralles, A., Huchard, M., Nebut, C.: Analyse de l'évolution d'un modèle : vers une méthode basée sur l'analyse formelle de concepts. In: XXIXème Congrès INFORSID (2011)
6. Falleri, J.R.: Contributions à l'IDM : reconstruction et alignement de modèles de classes. Ph.D. thesis, Université Montpellier 2 (2009)
7. Rouane, M.H., Dao, M., Huchard, M., Valtchev, P.: Aspects de la réinginierie des modèles uml par analyse de données relationnelles. Ingénierie des Systèmes d'information (RSTI série) **12**, 39–68 (2007)

8. Godin, R., Mili, H.: Building and maintaining analysis-level class hierarchies using galois lattices. In: Eighth Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), pp. 394–410 (1993)
9. Dao, M., Huchard, M., Hacene, M.R., Roume, C., Valtchev, P.: Towards practical tools for mining abstractions in uml models. In: International Conference on Enterprise Information Systems: Databases and Information Systems Integration (ICEIS 2006), pp. 276–283 (2006)
10. Arévalo, G., Falleri, J.-R., Huchard, M., Nebut, C.: Building abstractions in class models: Formal concept analysis in a model-driven approach. In: Wang, J., Whittle, J., Harel, D., Reggio, G. (eds.) MoDELS 2006. LNCS, vol. 4199, pp. 513–527. Springer, Heidelberg (2006)
11. Altmanninger, K., Seidl, M., Wimmer, M.: A survey on model versioning approaches. Int. J. Web Inf. Syst. **5**, 271–304 (2009)
12. Altmanninger, K., Schwinger, W., Kotsis, G.: Semantics for accurate conflict detection in smover: Specification, detection and presentation by example. Int. J. Enterp. Inf. Syst. **6**, 68–84 (2010)
13. Cicchetti, A., Di Ruscio, D., Pierantonio, A.: Managing model conflicts in distributed development. In: Czarnecki, K., Ober, I., Bruel, J.-M., Uhl, A., Völter, M. (eds.) MODELS 2008. LNCS, vol. 5301, pp. 311–325. Springer, Heidelberg (2008)
14. Cicchetti, A., Ruscio, D.D., Pierantonio, A.: A metamodel independent approach to difference representation. J. Object Technol. **6**, 165–185 (2007)
15. Ohst, D., Welle, M., Kelter, U.: Differences between versions of uml diagrams. SIGSOFT Softw. Eng. Notes **28**, 227–236 (2003)
16. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. In: Semantic Interoperability and Integration (2005)
17. Bendaoud, R., Napoli, A., Toussaint, Y.: Formal Concept Analysis: A unified framework for building and refining ontologies. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 156–171. Springer, Heidelberg (2008)
18. Stumme, G., Maedche, A.: Ontology merging for federated ontologies on the semantic web. In: International Workshop for Foundations of Models for Information, Integration (FMII-2001), pp. 413–418 (2001)
19. Formica, A.: Ontology-based concept similarity in Formal Concept Analysis. Inf. Sci. **176**, 2624–2641 (2006)
20. Tatsiopoulos, C., Boutsinas, B.: Ontology mapping based on association rule mining. In: International Conference on Enterprise Information Systems: Databases and Information Systems Integration (ICEIS 2009), pp. 33–40 (2009)
21. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. **18**, 323–364 (1986)
22. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J **10**, 334–350 (2001)
23. Shvaiko, P., Euzenat, J.: A Survey of Schema-Based Matching Approaches. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV. LNCS, vol. 3730, pp. 146–171. Springer, Heidelberg (2005)
24. Parent, C., Spaccapietra, S.: Issues and approaches of database integration. Commun. ACM **41**, 166–178 (1998)
25. Doan, A., Halevy, A.Y.: Semantic integration research in the database community: A brief survey. AI Mag. **26**, 83–94 (2005)

26. Miralles, A., Gorretta, N., Miller, P.C., Walklate, P., Van Zuydam, R.P., Porskamp, H.A., Ganzelmeier, H., Rietz, S., Ade, G., Balsari, P., Vannucci, D., Planas, S.: Orchard sprayers:an european program to compare testing methods. In: International Symposium on Fruit Nut and Vegetable Production Production Engineering, Valencia Zaragoza, ESP, 22–26 mars 1993, pp. 117–122 (1994)
27. Miralles, A., Polvêche, V.: Effects of the agrochemical products and adjuvants on spray quality and drift potential. In: 5th International Symposium on Adjuvants for Agrochemicals - ISAA '98, Vol. 1, Memphis (USA), pp. 426–432 (1998)

Advanced Product Lifecycle Management by Introducing Domain-Specific Service Buses

Stefan Silcher[1(✉)], Max Dinkelmann[2], Jorge Minguez[1], and Bernhard Mitschang[1]

[1] Graduate School of Excellence Advanced Manufacturing Engineering (GSaME), 70569 Stuttgart, Germany
stefan.silcher@gsame.uni-stuttgart.de
http://www.gsame.uni-stuttgart.de

[2] Institute of Industrial Manufacturing and Management (IFF), 70569 Stuttgart, Germany

Abstract. Manufacturing companies are operating today in a turbulent market. Permanently changing preconditions forces the companies to continuously adapt their business and production processes to get the optimal productivity. Therefore, a vast number of IT systems are introduced to support tasks along the product life cycle. These systems are typically isolated and their communication, cooperation and in special cases also integration results in more and more overhead and gets quickly unmanageable. Further problems arise, when building continuous processes within the Product Lifecycle Management (PLM). The service-based PLM architecture faces these challenges and presents a homogeneous integration approach based on Enterprise Service Bus (ESB) technology. The characteristics and findings of our approach are presented and the inclusion of security features is discussed. A proof-of-concept for the production planning and the corresponding Production Planning Service Bus are presented. Finally, the advantages of the service-based approach compared to traditional integration solutions are pointed out.

Keywords: Product lifecycle management · Service-oriented architecture · Enterprise service bus · Modular IT integration

1 Introduction

Manufacturing companies face several challenges nowadays. Firstly, in some industry sectors there exists an overcapacity of produced goods, for instance in the automobile industry. There is a tough competition between companies for market share, price and quality of products. Secondly, the trend for customization is growing. Companies deliver highly individualized products to customers in order to increase their competitiveness. This results in an increasing complexity in the development of products and production planning. Finally, environmental regulations, like reducing energy consumption and greenhouse gas emissions or replacement of harmful materials, force producing companies to be highly innovative in developing new technologies, materials and processes.

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 92–107, 2013.
DOI: 10.1007/978-3-642-40654-6_6,

These challenges lead together with a highly volatile market to new requirements for manufacturing companies. To consolidate or increase the market share, the adaptability of companies has to be enhanced in many ways. The production has to be as flexible as possible to produce highly customizable products in the same production line. Changes in the factory layout have to be performed in a very short time and often with increasing frequency in order to achieve shorter lead times. Therefore, the exchange of data between product development, the digital and physical factory has to be improved. This means that the actual state of the physical factory should be reflected in the model of the digital factory at any time. This enables a faster reaction on appearing problems or failures in the physical factory. Additionally, the organization has to be adaptive as well. Hence, changes in the business processes have to be smoothly performed [1].

The first approaches to organize and provide the product data over the whole life cycle came with the Product Data Management (PDM). Introducing PDM systems enormously reduces the effort to handle product data, but the PDM systems do not consider the processes in the whole product life cycle. Therefore, the concept of PDM was extended to monitor and manage these processes and Product Lifecycle Management was introduced (PLM). The goal is to optimize and standardize the processes to execute them efficiently and therefore save time and money [2,11].

Additionally, the tasks within a process are supported by software applications. This leads to a faster execution of single tasks and consequently reduces the execution time of the process. However, several problems arise when a high number of applications are installed. One of these problems is the emergence of information silos. Such an IT landscape can often be characterized as distributed, heterogeneous and proprietary. Coupling applications by implementing point-to-point interfaces reduces the problem of information silos, but such solutions get quickly very complex and hard to manage and maintain.

The motivation of our work is to build an architecture, which provides the needed flexibility in the IT landscape for manufacturing companies. The processes within PLM have to be continuously supported by IT systems. These systems have to be loosely coupled to provide the needed flexibility. Existing applications should be integrated into the new architecture. Additionally, the management and maintenance effort of the infrastructure has to be reduced and the availability of data within the product life cycle improved.

To realize such an architecture, a flexible solution is needed to integrate various heterogeneous applications throughout the product life cycle. A commonly used paradigm today is the Service-oriented Architecture (SOA). SOA provides a flexible integration of applications by loose coupling and reusing services, which are self-contained, platform-independent and discoverable [3]. The use of standardized technology like Web services to implement services allows the easy maintenance, extension or exchange of an interface. Moreover, Web services can be composed to workflows, which are modeled in standardized languages like Business Process Execution Language (BPEL) [4] or Business Process Model and Notation (BPMN) [5]. These workflows can be executed by corresponding workflow engines to support the business processes.

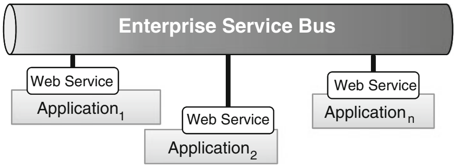


Fig. 1. Enterprise Service Bus.

The flexible composition of Web services within workflows enables a flexible IT support of business processes, which is necessary to quickly adapt the business processes in a highly volatile environment. The platform-independence of Web services allows exchanging data and information between heterogeneous applications. Relying on standards and loosely-coupled applications simplifies maintaining, changing and extending the IT infrastructure [6].

A common middleware solution to integrate Web services is the Enterprise Service Bus (ESB) [7], shown in Fig. 1. The ESB handles the routing of messages to enable a loose coupling of applications. Most ESB solutions possess a BPEL engine, which is able to execute BPEL workflows. Additionally, the ESB can offer functionality like message queuing capabilities or monitoring services.

This paper presents a service-based integration approach for PLM. The developed architecture is based on an ESB hierarchy to integrate the product life cycle in a modular way. The results of the implemented prototype for the digital factory demonstrates the benefits of the service-based solution. The benefits include the flexible composition of IT tasks, implemented as Web services, in workflows. The workflows support the planner of the factory by automating recurring tasks and saves therefore time and money. Additionally, the management and extensibility effort of the implemented prototype is enormously enhanced compared to the previous solution with customized scripts.

The remainder of the paper is structured as follows: In Sect. 2, the challenges in PLM and the problems of inadequate integration of applications are discussed. Furthermore, the service-based approach for PLM integration is presented and the current solutions in application integration are described. The implemented prototype for the integration of production planning tools is depicted in Sect. 3. In Sect. 4, the benefits and problems as well as further extensions are discussed before related work is presented. Finally, a summary and outlook is given in Sect. 5.

2 Service-Oriented PLM Architecture

An efficient and effective IT support of PLM processes is one of the major challenges in today's manufacturing companies. Using IT systems, especially in the product development and production planning phase, can reduce the time-to-market of new products tremendously. At the same time failures and ramp-up-time of production are reduced as well as the quality of planning results is improved. Today, the time between deciding to develop a new product and its production start can be further reduced by a seamless integration of the various tools in the product life cycle. These IT systems are often legacy applications or information silos difficult to integrate. In order to enhance the efficiency of production, the challenge is to enable an efficient exchange of information within the plethora of heterogeneous and distributed IT systems used during the product life cycle.

The adoption of the SOA paradigm eases the needed information exchange between heterogeneous applications by using standardized interfaces like Web services and decoupling service functionality from its implementation. Hence, a modular integration of the various product life cycle phases is presented in Sect. 2.2. Modularity reduces the implementation and maintenance effort of the IT infrastructure. Data exchange between the Web service as well as their compositions is accomplished by an ESB. Workflows are capable of automatically executing recurring tasks within PLM and therefore save valuable time, as well as reducing errors.

In Sect. 2.1, the separation of the product life cycle in six phases, which is used in this paper, is presented. The service-based PLM architecture is described in Sect. 2.2 and possible integration of security and safety features are discussed in Sect. 2.3. The ideas of a service-based PLM integration to supply chains are extended in Sect. 2.4. Subsequently, the motivation for an improved data exchange between the digital and physical factory as well as the current state of integration at the Learning Factory's digital learning shell of the Institute of Industrial Manufacturing and Management (IFF) at the University of Stuttgart is presented in Sect. 2.5.

The architecture for the production planning and its prototypical implementation is given in Sect. 3. This prototype realizes a seamless service-based integration of systems in the production planning at the IFF and eases the data management thanks to automating the data exchange between the applications by implementing the process in a BPEL workflow.

2.1 Product Life Cycle

The product life cycle is heterogeneous in many ways. Therefore, it is split in different phases, each of them represent a characteristic activity. In Fig. 2, the different phases of the product life cycle can be seen. The separation of the product life cycle considers not only the different activities in each phase, but also the support with tools and the management of data.

In the first phase, the *concept* phase (C), ideas for new products are generated and market analyses are conducted. Therefore, wikis, blogs or social networks are used and often unstructured data has to be handled and interpreted.

The *design & development* phase (DD) is determined by developing and designing the various parts of the products. Tools like Computer-aided Design (CAD) and Product Data Management systems (PDM) manage the high volume of data within this phase.

In the next phase, the *production planning* (PP) is carried out, where process and resource data are added to the product data and linked among each other.

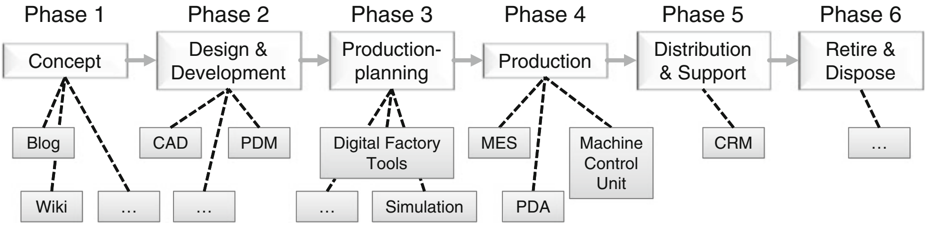


Fig. 2. Phases and Tools of the Product Life Cycle.

This data is linked in Digital Factory Tools, which generate a plan for the factory. The planned production processes are simulated with simulation tools to verify the sequence. The data volume, which has to be handled, increases enormously in this phase compared to the design and development phase.

Once the product development and production planning are completed, the production can start in phase four, the *production* phase (P). Here, Manufacturing Execution Systems (MES) are responsible for executing the production orders in the production. The control unit provides a real-time control of the manufacturing facilities. Feedback of the manufacturing facilities is gathered and stored by the production data acquisition unit (PDA) for later analysis.

Selling the products, their maintenance and the customers comprise are the main tasks of phase five, the *distribution* & *support* phase (DS). Customer Relationship Management Systems (CRM) are used for the linking of sold products and customers for any kind of complaints or warranty issues.

The *retire* & *dispose* phase (RD) deals with the disposal of products. Information about assembly and material composition of the product can help to regain valuable material or to simplify the separation of material for recycling.

2.2 Service-Oriented Integration

PLM extends the concept of PDM by adding management and control of business processes for the whole life cycle. The problem today is that IT systems poorly support business processes. Especially the flexible IT support of business processes is of great importance [8]. Therefore, a service-based architecture is needed, which allows to implement a continuous IT support along the whole product life cycle. Additionally, business processes must be adapted in an easy manner as part of the IT infrastructure.

Therefore, a modular service-based architecture was developed with different integration layers, to efficiently integrate the plethora of applications, used in the product life cycle, as well as to flexibly support the business processes. The developed architecture, which is presented in Fig. 3, uses a pillar for every phase of the product life cycle. Furthermore, the six phases are integrated with an additional pillar for the overall PLM [9].

Each pillar is represented by a phase-specific ESB that integrates all the applications of the corresponding phase. These phase-specific characteristics can

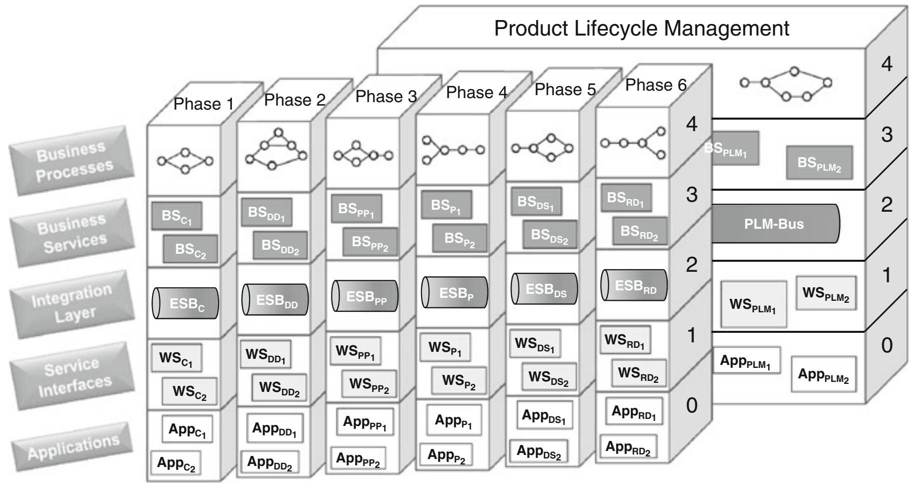


Fig. 3. Service-based Integration Architecture for Product Lifecycle Management.

be supported directly. A distinction is made between phase-specific ESBs and the phase-overlapping PLM-Bus. This solution with multiple ESBs builds a hierarchy to efficiently manage all the processes in the whole product life cycle.

The vertical integration of each product life cycle phase is clearly structured by distinguishing different levels of abstraction [10]. Hence, the functionality and data, provided by each application, is exposed as Web service in small, functional units. These units can be composed to workflows, which support the execution of business processes. The five layers are explained in Sect. 3.1 in more details.

The horizontal, phase-overlapping integration is performed by the PLM-Bus, which connects the phase-specific ESBs as a central backbone. The separation of phase-specific integration and holistic PLM integration contains several advantages [9]. Particularly, the possibility of adapting the ESB to the requirements of each phase like availability, data throughput and time requirements are of great importance.

The PLM-Bus is responsible to manage the phase-overlapping data exchange, tasks and processes. The PLM-Bus should provide information about available Web services of all phases as well as authorization and authentication as a single sign-on service. Processes like change and failure management have to be coordinated by the PLM-Bus. Additionally, Enterprise Resource Planning systems and other applications, which cannot be assigned to a specific phase, should be directly integrated by the PLM-Bus.

2.3 Security and Safety Issues in the Product Life Cycle

This subsection deals with security and safety issues in PML and how the presented service-based architecture can support these issues. Due to the high number of employees involved and vast quantity of different tasks executed in the

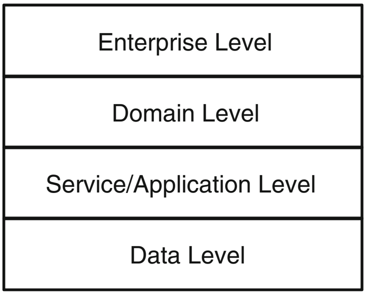


Fig. 4. Security levels in the service-oriented PLM architecture.

product life cycle, security and safety management should be naturally supported by the IT architecture. Therefore, a uniform authentication and authorization is needed, which can be provided by accordant Web services. Additionally, a rights and roles management has to be introduces, which restricts the access of each employee to the systems and data they need for their work. For this reason, different security levels were defined, which are presented in Fig. 4 and discussed in the following.

The objective is to securely maintain the product data [11]. The demand for a consistent rights management over the whole product life cycle, including authentication and authorization services, can be accomplished by the PLM-Bus, as the central instance in the service-oriented PLM architecture. This level of security belongs to the *Enterprise Level* of Fig. 4.

The other three security levels take place in the whole architecture, that means on the PLM as well as the phase-specific layers. The phase-specific integration of the product life cycle phases reflects an organizational separation of the IT infrastructure, which leads to a naturally defined security level, the *Domain Level*, because the field of activity of most employees is restricted to a single domain. Therefore, it is sufficient to give them admission to this domain, which can be further detailed on the *Service/Application Level*. The lowest and most detailed security level is described as *Data Level*, where access of users can be restricted to specific databases or other data sources.

Beside the access control other aspects like a secure communication of messages has to be considered. For the secure and reliable exchange of messages between Web services, the Organization for the Advancement of Structured Information Standards (OASIS) defined the WS-Security standard, which should be used to sign and encrypt the contend of each message [12].

To ensure data safety, three different approaches are conceivable to backup or replicate the enterprise data. Firstly, each application is responsible for the safety of its dedicated data. Secondly, a data backup is done centrally in each phase, by using a backup service deployed in each ESB. Finally, the safety of enterprise data can be accomplished by the PLM-Bus. This would implicate an uniform backup and replication strategy over the whole IT-infrastructure, but comes along with an increased number of messages to perform data safety.

2.4 Service-Based PLM Architecture Applied to Supply Chains

Previously not considered was the fact, that most companies are not involved in the whole product life cycle. The Original Equipment Manufacturers (OEMs) treat the product life cycle from the development to the recycling of the product. Small and Medium-sized Enterprises (SMEs), in particular suppliers of OEMs, often focus their work on one or two phases of the presented product life cycle. Therefore, a phase-specific integration is often sufficient to cover their whole business area. This is not only profitable for the SMEs, which benefit from the advantages of a service-based IT infrastructure. Especially for the OEMs, the change of the SMEs IT landscape to an SOA can be valuable.

A service-based communication between OEM and SME improves the quality of the data exchange, due to standardized Web service interfaces, workflow definition or data exchange formats. Additional, asynchronous communication eases the loose coupling systems of the companies.

Therefore, the integration of SMEs in the business process of an OEM is useful to have a defined way of communication between the companies. E.g. changes in the design of a product, made by an SME, can be necessary due to inconsistencies in assembly tests of the OEM. The OEM can describe the problems and send them to the SME, where automatically an exposed process is triggered to solve the resulting problem.

Therefore, the presented architecture can be easily extended to manage data and processes in the supply chain covering all involved companies, e.g., from OEM to all suppliers. The challenge would be to convince all suppliers and customers of a company to migrate their IT infrastructure to SOA. Beside the technical challenge it poses an organizational challenge.

2.5 Digital and Physical Factory

Today, one key to a more efficient factory is the coupling of the digital and physical factory. The vision is to have an up-to-date digital copy of the physical factory available at any time. Based on the current state, different simulations could be executed to forecast the short and medium term development of the factory and its production.

The problem is to provide status information of the production environment in real-time to the digital factory in order to automatically run a simulation model out of that data [13]. The presented architecture improves the availability of data in the production environment by exposing Web service interfaces and loosely coupling of applications in the infrastructure, which allows a faster data exchange between the digital and physical factory.

The IFF at the University of Stuttgart has built a Learning Factory, which contains a digital learning shell and a physical factory. The digital learning shell contains many tools for planning and optimizing the digital model of the factory. The central application is a database to store product, process and resource (PPR) data, also called PPR-Hub. Around this PPR-Hub, there are several tools for the factory optimization. Amongst others, there is a layout planning table, where a group of people can cooperatively optimize the factory layout [14]. In the next step, a logistic simulation tool is used to verify the new layout and planned production processes by simulating various scenarios [15].

Currently, these tools and others can exchange data with the PPR-Hub by executing (Visual Basic) scripts. These scripts are customized point-to-point connections. Therefore, the inclusion of new tools or substitution of an existent tool is associated with a great effort. Substituting a tool leads to a complete reimplementation of the script.

The physical factory of the IFF, called iTRAME, consists of modular manufacturing units, which are connected with a universal plug-and-play mechanism and can be easily exchanged [16].

The factory layout of the physical factory can already be automatically detected and copied with a script to the digital planning environment. Due to the high effort for implementing these scripts, no further information is used in the planning environment, e.g. process data and time information of the production. This information would help to understand the current situation in the production, when planning and real data are compared, and thus would enhance the planning accuracy, but also increases the planning effort.

The goal of implementing an SOA is to replace the tight coupling of the applications with scripts by Web service interfaces, which can be flexibly composed into workflows. The workflows can support the planner by automatically or semi-automatically executing recurring tasks in the product life cycle, e.g. when new products are introduced, changes in the product mix are detected, or machine failures appear.

3 Implementation of a Service-Based Integration for Production Planning

This section describes, how the service-based architecture is applied to the production planning phase. Subsequently, the implementation details of the prototype are described, where two applications of the Learning Factory's digital learning shell were integrated by the presented approach. Therefore, the databases of each application were equipped with Web service interfaces and a web-based portal was developed to control and manage workflows executed in the production planning environment. Additionally, the implemented integration workflow is presented.

3.1 Production Planning

The implementation of the presented service-based architecture for PLM in the production planning is illustrated in Fig. 5.

Layer 0 contains the various applications of the production planning, e.g. digital factory tools, layout planning table, simulation tools. The data and functionality of the applications is exposed to other tools by means of Web service interfaces, which are placed in Layer 1. The ESB in the integration layer (Layer 2)

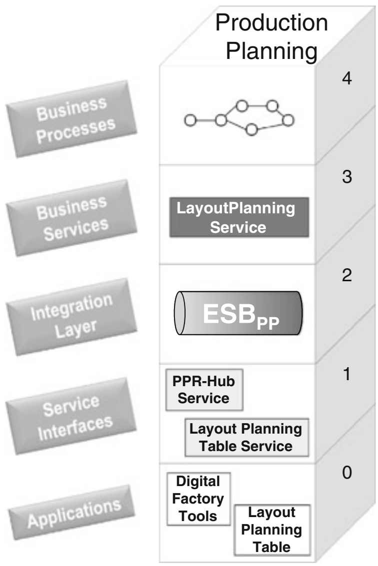


Fig. 5. Integration Pillar for the Production Planning Phase.

routes the messages sent and received by the Web service interfaces to the desired destination. Additionally, it performs the data transformations from proprietary data formats to a common message format of the ESB. The business services in Layer 3 compose the Web services to small processes, which manage the data exchange between two applications. The business process layer at the top of the pillar (Layer 4) represents the IT implementation of the business processes.

Web service technologies are platform independent, which is a great advantage when integrating proprietary heterogeneous systems like the current IT infrastructure at the IFF. In the implemented prototype, the messages are exchanged over Message Queues (MQ), which allow a reliable, asynchronous communication between the participating applications and improve the loose coupling of applications [17]. More details on our prototype implementation will be given in the subsequent section.

3.2 Example of Service-Based Production Planning

In the first phase of the implementation, the central data hub, which stores the PPR data, should be integrated with a layout planning tool [14]. The developed architecture is shown in Fig. 6.

The core of this architecture is the Production Planning Service Bus (PPSB), which is based on the OpenESB, an open source implementation of an ESB [18]. BPEL workflows can be executed in the workflow engine of the OpenESB. To administrate the workflows, the Workflow Management Portal was developed, where the production planner can start, control and if necessary restart or stop

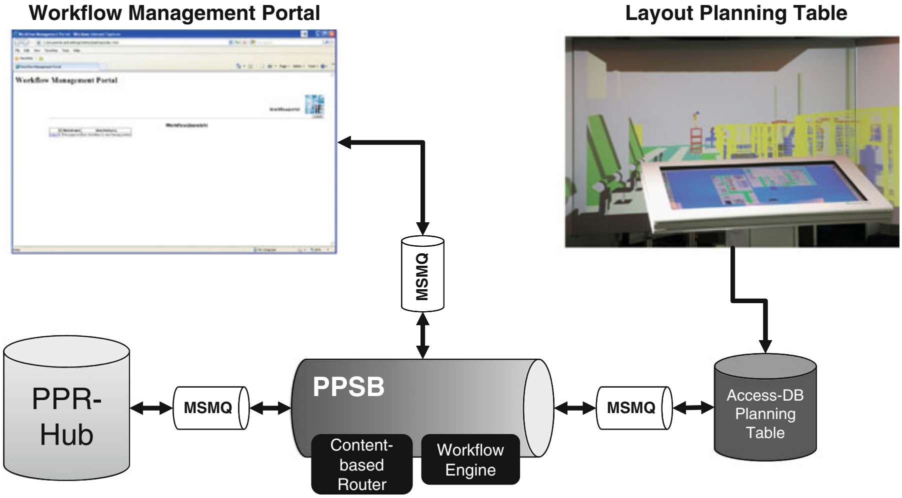


Fig. 6. Architecture of Integrated IT systems.

the available workflows. The usage of the implemented planning workflow is described in Sect. 3.3.

Using asynchronous communication improves the loose coupling between the applications. They do not communicate directly with each other; instead the messages are sent over MQs. An MQ enables a reliable communication between the participants by receiving messages and storing them persistently until the MQ has successfully delivered the message to the receiver. This way, the receiver can be temporarily unavailable without disrupting the whole system or blocking the sender of a message. Therefore, the usage of MQs makes the system more robust against network or application failures and improves the loose coupling of the applications.

In the presented prototype, Microsoft Message Queuing (MSMQ) is used to connect the systems with the OpenESB. To enable the OpenESB to communicate with the MSMQs, an MSMQ Binding Component is provided by the developers of the OpenESB.

To integrate the applications within the architecture, Web service interfaces were developed to enable the exchange of messages with other applications. For the Workflow Management Portal, PPR-Hub and Layout Planning Table a Web service interface was implemented for each. Due to the fact that no source code was available for the Layout Planning Table, the only way to exchange data with this tool was an interface to its Microsoft Access database.

All the messages exchanged in the system rely on the same common message format. Thus, all the messages have the same structure and don't have to be transformed between the different systems. The only effort is to generate a correct message, including the data provided by the data source. To reduce the effort, a common library that automatically generates the messages in the correct format is implemented to be used by all Web services.

The loose coupling of the integrated applications is ensured by using a content-based router [17]. This means, an application must not know the network address or endpoint of the destination system of a message. The message can just be sent to the ESB, where the content-based router determines the destination by inspecting the content of the message. Therefore, the endpoint can be looked up in a database, when the destination system is known. This allows transferring applications to other servers or changing their endpoints without affecting other applications. The only thing to do is to change the corresponding endpoint in the database of the content-based router and all other applications can communicate again with the changed application. The purpose of the content-based router is to decouple applications and to enhance adaptability.

The presented approach to integrate applications of the production planning allows the production planner to manage the data of the integrated applications at a single point, the Workflow Management Portal. In the portal, the planner can see the available workflows and start, stop or restart them. The portal can be easily extended to include new workflows, when more application are integrated into the infrastructure.

3.3 Layout Planning Workflow

The workflow implemented for the prototype controls the data flow between the systems presented in the previous subsection. A BPMN model of this workflow is presented in Fig. 7.

The first task *Load Projects* of the workflow sends all projects, which are stored in the PPR-Hub, in a message to the Workflow Management Portal. The message is extracted and the projects are presented in the portal.

The user is asked in the second task *Select Project*, to select the project he wants to modify in the Layout Planning Table. The human icon in the top left corner of the task indicates that a human interaction is necessary in this task.

After selecting a project, the third task *Load Resource Data to Planning Table* is started, which sends a message containing the selected project to the PPR-Hub. The resource information of this project is thereupon sent over the ESB to the Layout Planning Table and stored in its database.

In the fourth task *Perform Layout Planning*, the user can perform the layout planning to optimize the material flow, the logistic processes, and so forth. The hand icon in the top left corner of the task indicates that this task has to be performed manually by the user.

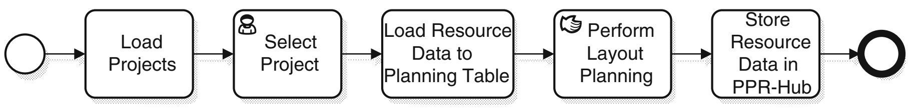


Fig. 7. BPMN Workflow for Layout Planning.

When the user finished the layout planning, the last task *Store Resource Data in PPR-Hub* is executed. This task generates a message to send the optimized planning data over the ESB back to the PPR-Hub, where they are stored and are now available for other systems in the production planning.

4 Review of the Approach and Related Work

The following section summarizes the benefits and discusses shortcoming of the presented approach. Finally, related work is given and compared with the service-based architecture.

4.1 Benefits of the Implemented Prototype

For the manageability of our IT infrastructure approach, this prototype demonstrates great advantages compared to the previously implemented point-to-point interfaces between the applications:

- The ESB as central integration backbone eases the connection of the heterogeneous applications to this prototype.
- The use of a common message format reduces the number of different transformations, which leads to a better extensibility and scalability of the infrastructure.
- The content-based router enables the loose coupling of the applications to the ESB by introducing a central database for the registration of application endpoints.
- The implementation of MQs boosts the loose coupling at the connectivity layer. Additionally, the robustness of the complete infrastructure is improved against temporary failures of networks or applications.
- Changing from a synchronous to an asynchronous communication increases the performance by reducing unnecessary blocking of applications when waiting till a message is sent or is available to be received.
- The platform-independent Web services technology enables unified interfaces to heterogeneous applications and databases as well as their loose coupling in workflows.

Proprietary applications can be integrated in various ways into an SOA. To access the functionality of a program, an available interface can be used or a new one can be implemented, provided that the source code is available. If neither is available, the functionality of a program cannot be easily integrated in a workflow, as in our case. Nevertheless, the data can be accessed by an interface over the program logic or directly over the database of a program. In the prototype the databases of the integrated programs were equipped with an Web service interface which lead to an enhances data exchange between the applications. To fully automate the planning processes, functionality has to be exposed as Web services to be able to execute them within a workflow activity.

Extending the prototype with a simulation tool like the logistic simulation tool would make sense. The optimized layout could be verified by simulating the production processes and the throughput can be measured. Therefore, the simulation tool has to be equipped with a Web service interface to receive the necessary data for the simulation.

The common message format has to be extended to include besides the resource data also product and process data. In the currently used common message format, this extension is already provided and can be easily performed. Additionally, the Web service interface of the PPR-Hub has to be extended to read and write the product and process data. On the other hand, the Web service interface of the Layout Planning Table remains unchanged. Two alternatives are possible, when integrating a simulation tool with a workflow. The presented layout planning workflow can be extended to include the simulation tool or a new simulation workflow can be implemented to control the data exchange between the PPR-Hub and the simulation tool. In the second case, the layout planning workflow and the new simulation workflow have to be executed consecutively.

4.2 Related Work

In the last few years, the main PLM vendors like Dassault Systèmes, PTC and Siemens PLM Software extended their PLM solutions with a service-based approach to get the desired continuous integration of the product life cycle [19]. However, they adopted their own proprietary integration middleware and thus created restricted interoperability properties: they lack possibilities to integrate systems of other vendors, miss flexibility in business process support and applications are not loosely coupled to the integration middleware. Furthermore, the interfaces are not open, so it is hard or even impossible for other software vendors to connect their applications to these middleware systems.

Rantzau et al. implemented a Data Change Propagation System called CHAMPAGNE for heterogeneous information systems [20]. The CHAMPAGNE platform manages dependencies between the schemas of different distributed applications. Compared to the presented prototype in this paper, CHAMPAGNE implements a tight coupling to the participating applications. Hence, each change in a coupled system lead to changes in the source code of corresponding propagation scripts, which have to be performed manually by a software developer or an administrator.

5 Conclusions and Outlook

Highly volatile markets and growing competition force companies to continuously increase their effectiveness. The tough competition, growing customization of products and environmental regulations forces companies to continuously adapt their business processes. The flexibility of the company has to be improved to adapt to the constantly changing environment. This can be achieved by a more flexible support of business processes and the IT infrastructure. Additionally, the applications, which support a business process task, have to be better integrated to improve the data and information flow. The vision is a continuous integration of all applications used during the product life cycle to accelerate the data exchange.

The paper presents a service-based architecture to integrate the different phases of the product life cycle. The phase-overlapping integration is performed by the PLM-Bus, which allows exchanging data and coordinating processes between the phases. The benefits of this architecture are a clear separation between different levels of abstraction as well as the possibility to adapt each ESB to the requirements of its phase such as availability, data throughput and time requirements. Additionally, the integration of security issues in this architecture is discussed.

Furthermore, the developed prototype based on the Production Planning Service Bus performs a service-based integration of the production planning environment at the IFF. The benefits and problems of the prototype and the integration of proprietary applications are discussed, like a flexible composition of Web services in a BPEL process, and an outlook on useful extensions of the implementation in the production planning is given.

The next step in the service-based integration of PLM is the implementation of the PLM-Bus to efficiently couple the production planning and production phase. The goal is to establish a bidirectional communication between the digital and physical factory to automatically adopt the current production status for the planning and to accomplish an optimized planning in the production environment.

Acknowledgements. The authors extend their sincere thanks to Fabian Laux, who contributed in developing and implementing the prototype of the service-based production planning integration.

Furthermore, the authors would like to thank the German Research Foundation (DFG) for financial support of this project as part of the Graduate School of Excellence advanced Manufacturing Engineering (GSaME) at the University of Stuttgart.

References

1. Jovane, F., Westkämper, E., Williams, D.: The ManuFuture Road: Towards Competitive and Sustainable High-Adding-Value Manufacturing, 1st edn. Springer, Berlin (2009)
2. Saaksvuori, A., Immonen, A.: Product Lifecycle Management, 3rd edn. Springer, Berlin (2008)
3. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall International, New Jersey (2005)
4. OASIS: Web services business process execution language version 2.0 (2007)
5. OMG: Business process model and notation (BPMN) version 2.0 (2011)
6. Weerawarana, S., Curbera, F., Leymann, F., Ferguson, D.F., Storey, T.: Web Services Platform Architecture: Soap, WSDL, WS-Policy, WS-Addressing, WS-Bpel. WS-Reliable Messaging and More. Prentice Hall International, New Jersey (2005)

7. Chappell, D.A.: Enterprise Service Bus: Theory in Practice, 1st edn. O'Reilly Media, Sebastopol (2004)
8. Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: state of the art and research challenges. Computer **40**, 38–45 (2007)
9. Silcher, S., Minguez, J., Mitschang, B.: Adopting the manufacturing service bus in a service-based product lifecycle management architecture. In: Proceedings of the 44th International CIRP Conference on Manufacturing Systems: ICMS '11, Madison, Wisconsin, USA, Online, pp. 1–6 (2011)
10. Minguez, J., Ruthardt, F., Riffelmacher, P., Scheibler, T., Mitschang, B.: Service-based integration in event-driven manufacturing environments. In: Chiu, D.K.W., Bellatreche, L., Sasaki, H., Leung, H.-F., Cheung, S.-C., Hu, H., Shao, J. (eds.) WISE Workshops 2010. LNCS, vol. 6724, pp. 295–308. Springer, Heidelberg (2011)
11. Stark, J.: Product Lifecycle Management: 21st Century Paradigm for Product Realisation (Decision Engineering), 1st edn. Springer, Berlin (2004)
12. OASIS: Oasis web services security (WSS) tc (2006)
13. Kádár, B., Lengyel, A., Monostori, L., Suginishi, Y., Pfeiffer, A., Nonaka, Y.: Enhanced control of complex production structures by tight coupling of the digital and the physical worlds. CIRP Ann. Manufact. Technol. **59**, 437–440 (2010)
14. Kapp, R., Le Blond, J., Westkämper, E.: Fabrikstruktur und logistik integriert planen: Erweiterung eines kommerziellen werkzeugs der digitalen fabrik für den mittelstand. wt Werkstattstechnik, online 95, pp. 191–196 (2005) (German)
15. Kapp, R., Löffler, B., Wiendahl, H.H., Westkämper, E.: Der logistik-prüfstand: Skalierbare logistiksimulation von der lieferkette bis zum arbeitsgang. wt Werkstattstechnik, online 93, pp. 31–38 (2003) (German)
16. Dinkelmann, M., Riffelmacher, P., Westkämper, E.: Training concept and structure of the learning factory advanced industrial engineering. In: ElMaraghy, H.A. (ed.) Enabling Manufacturing Competitiveness and Economic Sustainability. Proceedings of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2011), pp. 624–629. Springer, Berlin (2011)
17. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Longman, Amsterdam (2003)
18. OpenESB: http://www.logicoy.com/esb.php (2010). Accessed Aug 2012
19. CIMdata: Service-oriented architecture for plm - an overview of ugs soa approach. Technical report, CIMdata, Inc. (2006)
20. Rantzau, R., Constantinescu, C., Heinkel, U., Meinecke, H.: Champagne: data change propagation for heterogeneous information systems. In: Proceedings of the 28th International Conference on Very Large Data Bases (VLDB '02), pp. 1099–1102, VLDB Endowment (2002)

Matcher Composition Methods for Automatic Schema Matching

Daniel Nikovski[1(✉)], Alan Esenther[1], Xiang Ye[1], Mitsuteru Shiba[2], and Shigenobu Takayama[3]

[1] Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA
nikovski@merl.com
[2] Mitsubishi Electric Corporation, 5-1-1 Ofuna, Kamakura, Kanagawa 247-8501, Japan
[3] Mitsubishi Electric Information Systems Corporation, 325 Kamimachiya, Kamakura, Kanagawa 247-0065, Japan

Abstract. We address the problem of automating the process of deciding whether two data schema elements match (that is, refer to the same actual object or concept), and propose several methods for combining evidence computed by multiple basic matchers. One class of methods uses Bayesian networks to account for the conditional dependency between the similarity values produced by individual matchers that use the same or similar information, so as to avoid overconfidence in match probability estimates and improve the accuracy of matching. Another class of methods relies on optimization switches that mitigate this dependency in a domain-independent manner. Experimental results under several testing protocols suggest that the matching accuracy of the Bayesian composite matchers can significantly exceed that of the individual component matchers, and the careful selection of optimization switches can improve matching accuracy even further.

Keywords: Data integration · Virtual databases · Uncertain schema matching

1 Introduction

It is often necessary to establish a correspondence (matching) between the schemas of two or more databases, that is, determine which schema elements refer to the same concept or physical object in the problem domain. This need arises in multiple tasks in the area of data management, such as data integration, data migration, and also the creation of virtual databases that expose the same unfiying data model while retrieving data from multiple physical databases. As a result, there has been significant research into automated and semi-automated methods for schema matching [1].

It is widely acknoledged that the automatic schema matching (ASM) problem is very difficult, because when database designers create database schemas,

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 108–123, 2013.
DOI: 10.1007/978-3-642-40654-6_7,

they rarely provide full and unambiguous information about what individual schema elements represent, and if any such information exists, it is usually not meant for computer processing. Rather, database designers usually choose suitable words or abbreviations for the names of data elements, so as to facilitate future maintenance of the data schemas by themselves or other humans. Lexical analysis of the names of data elements, then, is an important approach to ASM. For example, the names "Street", "Str", and "StreetName" can be recognized to refer to a street, possibly in an address, and lexical analysis by string matching can reveal this similarity. A different type of information that might be useful for ASM is the structure of the data schemas, if present. In many cases, schemas are not represented by a flat list of element names, but the elements are organized in a hierarchy. For example, the element "CustomerName" might have three subelements, "FirstName", "MiddleInitial", and "FamilyName". Using such structural information is another approach to ASM. Many more approaches exist, too. For example, when the actual values of two database fields come from the same statistical distribution (e.g., over names, numbers, etc.), this can serve as evidence that the corresponding schema elements match. Dictionaries, thesauri, and other auxiliary data sources have been used for ASM purposes, too [1].

Due to the difficulty of the problem, no single method has been shown to perform best on all ASM tasks. This has led to the idea that multiple basic matchers of the types described above can be used together in a composite matcher [2,10]. The purpose of the composite matcher is to combine the output of the individual matchers and arrive at a more accurate set of likely matches. In most cases, the output of an individual matcher k for a given pair of elements $S_1.E_i$ and $S_2.E_j$ is a similarity value v_k in the interval $[0, 1]$, where $v_k = 0$ means no similarity, and $v_k = 1$ means full confidence that the two elements match. When given a library of K different individual matchers, the objective, then, is to find a composite similarity measure v that is a function of the individual outputs v_k, $k = 1, K$.

Several methods for combining similarity values have been proposed. The LSD system [4] uses machine learning techniques to estimate weighting coefficients w_k such that the final similarity measure v is a weighted average of the individual similarity measures: $v = \sum_{k=1}^{K} w_k v_k$. The COMA system [2] extends this approach with the minimum and maximum operators: $v_{min} = \min_k w_k v_k$ and $v_{max} = \max_k w_k v_k$.

Although experimental results suggest that these methods for combining similarity values lead to matching accuracy that is higher than that of the accuracy of the individual matchers, it can be recognized that they are specific approaches to the fundamental problem of combining evidence from multiple sources (in this case, multiple individual matchers), and make very specific assumptions about the statistical structure of the evidence that might or might not be warranted in practice. In Sect. 2, we propose a general method for correct modeling of any kind of statistical structure in the evidence, based on Bayesian networks and probabilistic reasoning, and a statistically grounded method for composing matcher evidence using these Bayesian networks, and in Sect. 3, we describe the performance of the composite matcher on benchmark problems.

We also investigated the limits on the accuracy obtainable by means of matcher composition by analyzing the type of mistakes made by the basic matchers, and tailoring the combination methods to the kind of elements that were being matched. For example, we discovered that very different matchers were useful for matching leaf nodes and internal nodes in the schemas. Based on this analysis, we devised new composite matchers that were able to increase the matching accuracy even further, compared to the Bayesian approach, although it remains to be seen how these matchers would peform in new domains. These additional matchers are described in Sect. 4.

2 Bayesian Networks for Combining Outputs of Multiple Schema Matchers

When combining evidence from multiple sources, one of the major problems and causes for errors is the improper modeling of correlation and other forms of statistical dependence between variables in the problem domain. For example, when two very similar matchers k and l are applied to an ASM problem, their outputs v_k and v_l will be highly correlated — when v_k is high, then v_l will be high, too, and vice versa. For example, a lexical matcher based on edit (Levenshtein) distance would assign a medium-level similarity to the pair of element names "Street" and "State"; similarly, a lexical matcher based on the Jaccard distance between the sets of letters in the two elements would assign such similarity to the pair. For another pair of elements, for example "Street" and "Address1", both lexical matchers would compute low similarity. In either case, not only is the computed similarity misleading as regards to the correct match, but both matchers provide the same kind of evidence (both positive or both negative), so its (in this case, harmful) influence is reinforced. If a weighted sum of the two similarity values is used, the same evidence will be counted twice, in practice, which will result in a phenomenon known as over-confidence. One of the matchers is almost redundant, and including it in the composition process might actually decrease the accuracy of matching. This effect has been observed in other fields where evidence has to be combined, such as medical diagnosis, and one possible tool for handling it has been belief reasoning in Bayesian networks. Our method for combining matcher output is based on such a network.

2.1 Representation

A Bayesian network (BN) is a probabilistic graphical model that represents a set of random variables and their conditional dependencies by means of a directed acyclic graph (DAG). An edge in the DAG between two nodes signifies that the variable Y corresponding to the child node is statistically conditionally dependent on the variable X corresponding to the parent node. This dependence is expressed in a conditional probability table (CPT) stored in the child node for Y. If $X \in Par(Y)$, where $Par(Y)$ is the set of parent nodes of Y, this table contains probability entries $Pr(Y = y|Par(Y) = z)$ for every possible combination of values x that X can take on and configurations (sets of values) z that the variables in $Par(X)$ can take on. Likewise, when there is no direct edge between two nodes, they are assumed to be conditionally independent given their parents. In particular, when two nodes have a common parent, but no edge between them, they are assumed to be conditionally independent given the value of their parent. The presence (or absence) of edges in the DAG of a Bayesian network is a way to express the statistical dependence (correlation) between variables.

A Bayesian network to be used for combining outputs of individual matchers in an ASM task is shown in Fig. 1. Its DAG is a tree of depth four, with some additional edges between some of the nodes. The meaning of the nodes is as follows:

1. At the first (top) level, the root node corresponds to a Boolean variable signifying whether two schema elements match. This is the final hypothesis that has to be evaluated.
2. The nodes at the second level of the trees represent independent ways in which the two element names can match (lexical, structural, instance-based, etc.). It is expected that these variables are largely uncorrelated, because they use different information to test for possible matches. They also each correspond to clusters of individual matchers whose output is correlated. In Fig. 1, one cluster represents the hypothesis that the two elements match lexically, and

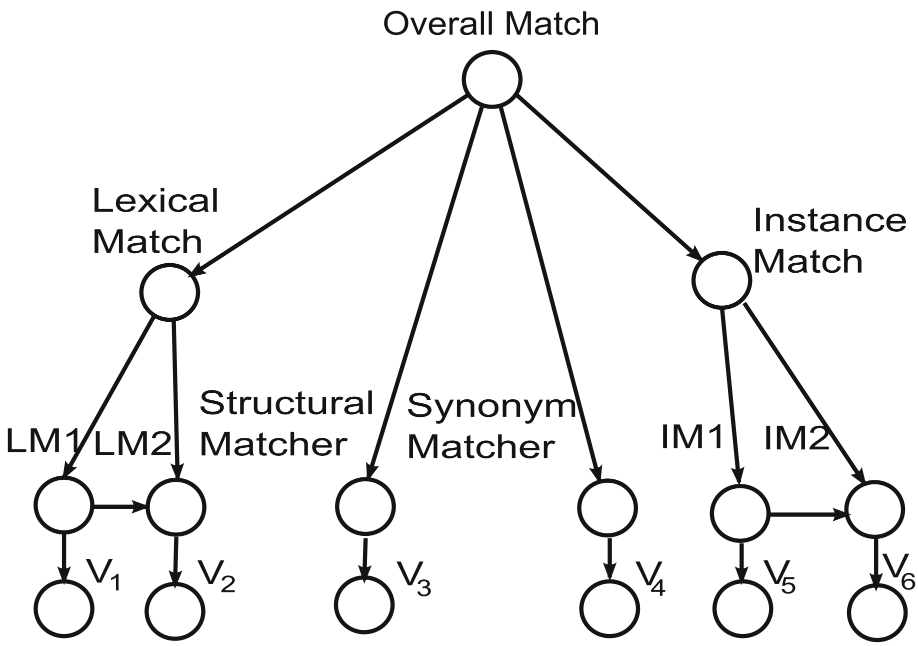


Fig. 1. A Bayesian network for combining the output of multiple individual matchers.

the other cluster represents the hypothesis that the instances (values) of the two elements in their respective databases match.

3. The nodes at the third level of the tree are also Boolean and represent the individual hypothesis that the two elements match, according to a single matcher. In Fig. 1, these include two lexical matchers LM1 and LM2, one structural matcher, one synonym matcher, and two instance matchers IM1 and IM2.
4. The leaves of the tree, at the fourth level, represent the similarity values V_k, $k = 1, K$ of the individual matchers whose outputs have to be combined (in this case, for illustration, $K = 6$). These variables are continuous, and their possible values are the real numbers v_k.

The overall structure of the BN expresses the understanding that when two elements match (or don't), the outputs of the structural matcher, synonym matcher, the lexical match variable, and the instance match variable will be statistically independent. This is what is to be expected on a matching task, because these matchers all use different information from the two data schemas in order to make an estimate about whether the elements match. However, the outputs of the two lexical matchers LM1 and LM2 would be correlated, as expected if they use the same information (the names of the two elements). That is why there exists an edge between nodes LM1 and LM2. Similarly, the output of the two instance matchers would be correlated, too, because they would both use the same information to base their estimates on (namely, the contents of the two corresponding database fields). Accordingly, an edge between nodes IM1 and IM2 reflects this dependency. This structure of the BN, then, corresponds to our understanding of which matchers produce highly correlated outputs, and which ones are statistically independent.

2.2 Parameter Estimation

In addition to the graph of the BN, if the network is to be used for inference, the parameters in its CPTs have to be specified, too. This can be done by means of labeled cases, where pairs $e_l = (S_1.E_i, S_2.E_j)$ of elements $S_1.E_i$ and $S_2.E_j$, $l = 1, \ldots, N$ have been run through all K matchers, to produce the corresponding similarity values $v_{l,k}$, $l = 1, \ldots, N$, $k = 1, \ldots, K$, and the correct labeling for some or all of the remaining Boolean variables has been supplied, too.

If labels for all Boolean variables have been supplied, then the estimation of the probabilities in the CPTs of the Boolean nodes could be reduced to frequency counting. That is, the entry $Pr(Y = y|Par(Y) = z)$ is equal to the ratio of the number of cases when Y had a specific value y (either True or False) and the parents of Y were in configuration z, and the number of times the parents of Y were in configuration z (regardless of the value of Y). For the continuous nodes V_k, a suitable parametric model for the similarity values must be chosen. One possible model is a normal (Gaussian) distribution with mean μ and variance σ^2. Then, two separate normal distributions $N(\mu_{k,+}, \sigma^2_{k,+})$ and $N(\mu_{k,-}, \sigma^2_{k,-})$ are estimated for positive (matching) and negative (non-matching) cases (pairs of elements), respectively. The mean $\mu_{k,+}$ is the average of the similarity values $v_{k,i}$ of all data cases where the parent node X_k of V_k has been labeled with value True. The parameter $\sigma_{k,+}$ is the sampled standard deviation of these cases. Analogously, the parameters $\mu_{k,-}$ and $\sigma_{k,-}$ are the sample mean and standard deviation of $v_{k,i}$ over all cases when the parent node X_k has been labeled with the value False.

It is also possible to estimate the parameters in the CPTs when only some of the nodes have been labeled. A typical situation arises when a human designer has provided feedback about whether the two elements match (that is, has assigned a Boolean value to the root node of the BN), but has not explained why they match (that is, whether the match is lexical, instance-based, structural, based on a dictionary, etc.) This situation is more challenging, but as long as the graph of the network is known and fixed, it is still possible to estimate the most likely values of the parameters in its CPT. This problem is known as parameter learning with partially observed data in Bayesian networks, and can be solved by means of gradient ascent in the likelihood function or the Expectation Maximization algorithm, among other methods [9,11].

Assuming there is a data set Σ of N independent training cases, the log-likelihood scoring function is

$$log\, L(\Theta|\Sigma) = \frac{1}{N}\sum_{i=1}^{M}\sum_{l=1}^{N} log\, P(X_{il}|Pa(X_i),\, \theta_i),$$

where Σ denotes the training data set, $Pa(X_i)$ denotes the parents of the node X_i, $i = 1, \ldots, M$, and Θ is the parameter vector $\Theta = \{\theta_1, \ldots, \theta_M\}$.

However, we only have partial observations, which means that there are several hidden nodes with no labels. For each training case, one pair of elements $S_1.E_i$ and $S_2.E_j$ is run through all K individual matchers to produce the corresponding similarity values $v_{i,j,k}$, and a true label of two elements matching or not for the root node $OverallMatch$ is provided by the human designer. With known structure and partial observation, we can use the EM (expectation maximization) algorithm to find a locally optimal maximum-likelihood estimate of the parameters. After learning parameters from training data set, each discrete node has a conditional probability table (CPT) specifying the probability of each state of the node given each possible combination of parents' states.

2.3 Inference

Given the individual similarity values $V_k = v_k$, $k = 1, K$ that have been reported by all individual matchers, and a full Bayesian network with CPTs estimated from data, we can evaluate the probability that the two elements match on the basis of all evidence, by means of a standard computational process known as belief updating. One possible method to perform belief updating is to construct the join tree of the Bayesian network, and use if for inference. This can be done by means of a number of commercial and freely available reasoning engines. The continuous variables V_k, under the chosen Gaussian parametrization, can be incorporated into the process of belief updating in the form of virtual (uncertain) evidence [12]. To supply virtual evidence to a belief updating engine, all that is needed is the likelihood ratio of the observed values v_k for the similarity value variables V_k:

$$L(V_k = v_k|X_k) \doteq \frac{Pr(V_k = v_k|X_k = T)}{Pr(V_k = v_k|X_k = F)} = \frac{N(v_k|\mu_{k,+}, \sigma^2_{k,+})}{N(v_k|\mu_{k,-}, \sigma^2_{k,-})},$$

where $N(v|\mu, \sigma^2)$ is the probability that measurement v comes from normal distribution with mean μ and variance σ^2, and X_k is the parent node of V_k in the BN.

After the process of belief updating concludes, all Boolean nodes in the network will be assigned probability values according to the observed evidence (values) v_k for the similarity value variables V_k. The probability of the root node is the final estimate that the two elements match, given the combined evidence of the individual matchers.

3 Experimental Results

In order to evaluate the match accuracy of any matcher described below, we used five XML schemas for purchase orders, CIDX, Excel, Noris, Paragon and Apertum, kindly provided to us by the University of Leipzig. The figure of merit for evaluation of the accuracy of matching was the popular f-measure, defined as the harmonic mean of precision and recall, as used in the information retrieval community. If the number of true matches identified by the matching system as such (hits) is A, the number of true matches not identified as such (misses) is B, and the number of cases when two elements do not match, but the matcher incorrectly declares a match (false positives) is C, the f-measure F can be computed as $F = 2A/(2A + B + C)$.

We developed 13 basic schema matchers and evaluated the ability of the proposed Bayesian method to combine their outputs so as to improve the accuracy of matching. Of these, 11 were lexical matchers: CosineSimilarity, HammingDistance, JaroMeasure, LevenshteinString, BigramDistance, Trigram Distance, QuadgramDistance, PrefixName, SuffixName, AffixName, Substring Distance. One matcher, PathName, was structural, comparing the entire paths of the two elements in their respective XML schemas. The last basic matcher was neither lexical nor structural: the Synonym matcher declared a match if and only if the two tested elements were found in a list of synonyms relevant to the domain of purchase orders. Based on their method of operation, the similarity values computed by the 11 lexical matchers can be expected to be highly correlated and statistically dependent; in contrast, the synonym matcher could be expected to produce output that is largely independent of the lexical matchers. Experimental evaluation of their pairwise dependence confirms this intuition: Figure 2 shows the pairwise correlation between all 13 pairs of matchers, evaluated from all pairs of elements in all ten pairs of schemas. Clearly, all 11 lexical matchers are highly correlated, whereas their correlation with the Synonym

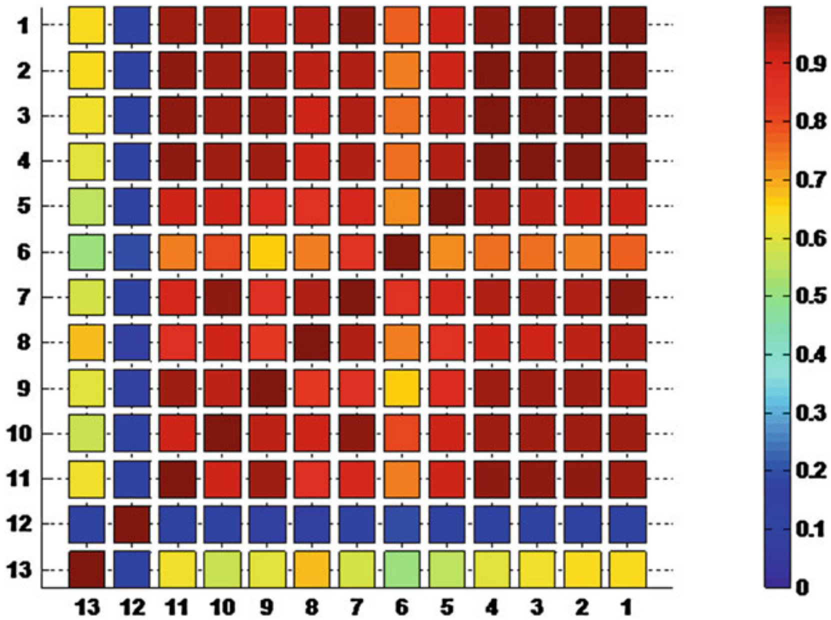


Fig. 2. Pair-wise correlations between all pairs of basic matchers, numbered as follows: 1: Edit Distance; 2: Sub-string Distance; 3: Bi-Gram Distance; 4: Tri-Gram Distance; 5: Quad-Gram Distance; 6: Cosine Similarity; 7: Hamming Distance; 8: Jaro Measure; 9: Affix Name; 10: Prefix Name; 11: Suffix Name; 12: Path Name; 13: Synonym.

matcher is minimal. Somewhat surprisingly, the structural matcher, PathName, is the least correlated with any other matcher.

The kind of major correlation that exists between lexical matchers is illustrated in Fig. 3 that shows a scatter plot of the similarity values computed by the Edit (Levenshtein) Distance matcher and the Sub-string Distance matcher. Their high correlation (0.9892) makes one of them almost redundant, if the other one is present.

Regarding the experimental evaluation of matching accuracy, as with any machine learning method, care should be given to the training and testing evaluation protocol, that is, which data are used for training and which data are used for testing. We used three evaluation protocols, as described below.

3.1 Testing on Training Data Set

This is the simplest evaluation protocol, where we use the same data set for testing and training. Its purpose is to evaluate how well we can fit the training data. Under this protocol, we define ten matching tasks that correspond to all possible pairs of the five XML schemas. For each matching task (pair of schemas), we build a dedicated Bayesian composite matcher that is specific for this task. The same data set, then, is used as evidence to predict the belief for every

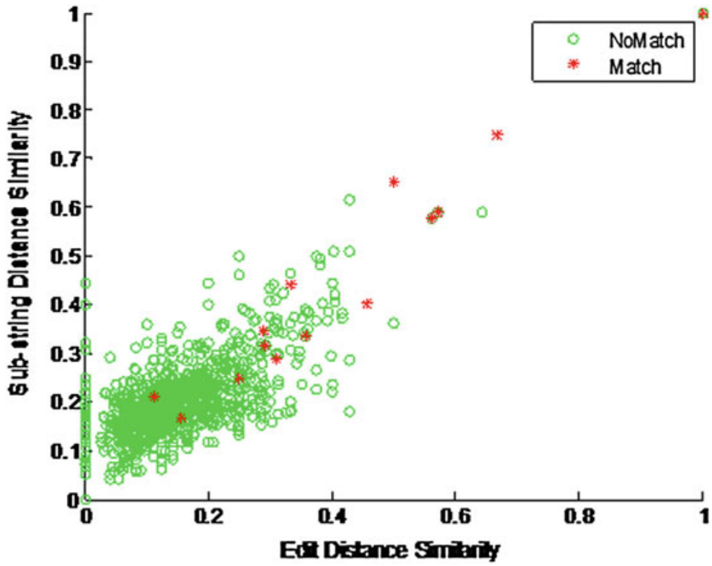


Fig. 3. Scatter plot of similarity values computed by the Edit Distance and Sub-string Distance matchers. Their output is clearly correlated, resulting in a correlation coefficient of 0.9892.

pair of elements. This is the most lenient evaluation protocol, since the learning algorithm has seen during training the data that will be used for testing.

After a similarity matrix is computed for all pairs of elements of two database schemas, an additional global matching step called Max1/Delta is performed to produce the final match decisions, based on the understanding that most often (but not always) mappings between database elements are one-to-one [2]. Since this procedure is sensitive to the exact value of the Delta parameter, we present below results as a function of it. After global match decisions have been obtained, they are compared with the ground truth, and the f-measure for this pair of schemas is computed. These f-measures are averaged over all pairs of tasks in the testing data set (in this case, ten pairs of tasks), in order to arrive at the final overall f-measure.

Figure 4 shows a comparison between all 13 basic matchers and the Bayesian Composite Matcher (BCM). The accuracy of the BCM reaches 0.819 and is significantly higher than that of any other matcher. It is also practically constant for a wide range of the parameter Delta. The performance of Path Name matcher is better than other individual matchers, because it is a hybrid matcher combining two basic match techniques.

3.2 Leave-One-Out Cross Validation (LOOCV)

A more realistic testing protocol is under the leave-one-out cross validation (LOOCV) method, where training and testing data are clearly separated. Each

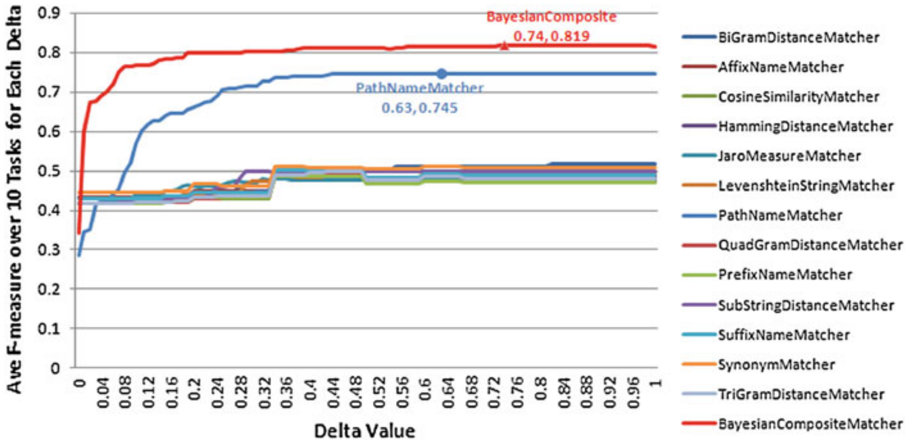


Fig. 4. Comparison of average f-measure between the Bayesian Composite Matcher and all other matchers.

of the ten pairs of schemas is used for testing, using a BCM that was learned using the other nine pairs of schemas. The results are averaged over the ten pairs, as follows:

1. Build training and testing data sets for 10 test tasks. For instance, if the similarity matrix of $Excel \leftrightarrow Noris$ is used as testing set, the training data set for this test task is a collection of similarity matrices of the remaining 9 schema pairs.
2. Learn one Bayesian composite matcher for each task based on its training data.
3. Implement $Max1/Delta$ selection approach on the composite similarity matrix generated by each Bayesian Composite Matcher.

3.3 Exclusive Leave-One-Out Cross Validation (ExclLOOCV)

The second protocol described above still allowed the training algorithm to see data from the pair of schemas that would be used for testing, but not the ground truth for their direct match. To eliminate any exposure of the training algorithm to data that would be used for testing, we modified the LOOCV procedure as follows. For each task, if the test pair is $A \leftrightarrow B$, the training examples only come from the three remaining schemas not involving either A nor B. For example, if one test set is $Excel \leftrightarrow Noris$, it will be tested with the Bayesian composite matcher that has used only the following three pairs of schemas for training: $CIDX \leftrightarrow Apertum$, $CIDX \leftrightarrow Paragon$, and $Apertum \leftrightarrow Paragon$. This is the maximally realistic testing protocol.

Figure 5 shows a comparison between the two variants of the LOOCV evaluation protocol for the Bayesian Composite Matcher. It can be seen that the accuracy drops to 0.76 under usual LOOCV and 0.73 under exclusive LOOCV.

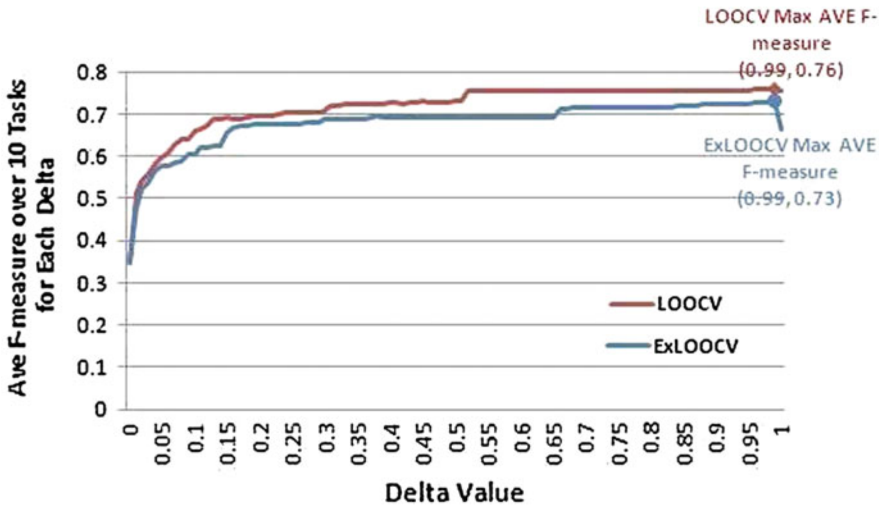


Fig. 5. Comparison of Bayesian composite matcher performance under LOOCV and exclusive LOOCV testing protocols.

4 Non-Bayesian Matcher Composition Optimizations

Based on familiarity with the domain acquired during the Bayesian Networks approach, we further explored non-Bayesian matcher composition approaches. A series of optimizations proved promising and are described here.

Typically, datasets have element names with strong lexical components, so it was critical to develop a sound fundamental lexical (name) matcher that could be heavily used as a component in higher-level composite matchers. The resulting Name matcher was a combination of fundamental matchers described in Sect. 3, using token processing and largely COMA-style techniques [2] on the element name tokens. Incorporation of abbreviation and synonym information was an integral part of this process, too. As an example, averaged across all 10 schema pairs in the COMA dataset, we only achieve an f-measure of 0.433 using a simple prefix name matcher, which looks for identical prefixes in column names. A synonym matcher, utilizing a lookup table is also an important component, but only achieved an f-measure of 0.461 by itself. Through experimental analysis, we found that combining lexical matchers into a Name Composite matcher achieved a marginal increase to 0.494, but that the resulting matcher was a critical component of higher-level matchers. A PathName matcher, which compares the full paths to each element in their XML schemas, based on this Name matcher boosted the f-measure to 0.820. Note that these matchers incorporated additional optimizations that were applied in parallel, some of which are described below.

The next step was to develop a Structure matcher which uses different strategies based on the type of nodes (root, interior or leaf node) being compared [19].

Table 1. Best matchers for various combinations of node types.

XML Schema node type	Matcher applied
LEAF-LEAF	PathName
ROOT-ROOT	LeafPath
INTERIOR-INTERIOR	ChildPath, SiblingPath
INTERIOR-ROOT	ChildPath
INTERIOR-LEAF	LeafPath
ROOT-LEAF	LeafPath

The matchers that we found to produce the best overall f-measures are shown in Table 1.

Here LeafPath is an application of the Name matcher to the paths to all leaves of the element nodes being compared. Similarly, ChildPath is an application of the Name matcher to the paths to all of the children of the nodes begin compared, and SiblingPath is an application of the Name matcher to all of the paths to the siblings of the nodes being compared. For the ten schema pairs in the COMA dataset, the Structure matcher achieved an average f-measure of 0.871. In addition to combining matchers, it was often found to be useful to "pre-filter" candidate element pairs by simply eliminating those for which the Name similarity was less than 50 % This REQUIRE_ELEMENT_NAMES_TO_BE_SIMILAR optimization was originally created as an optimization to improve compute time, but actually had a noticeable positive effect on overall results.

We next developed a LinearComposite matcher which allows one to manually specify weights for component matchers. While this technique itself is by no means automatic, there were some interesting findings in this work. The matcher was initially developed as a mechanism to strengthen effects of the Name and SiblingPath matchers, which the StructureMatcher seemed to diminish in some cases. The LinearComposite matcher was enhanced to consider the types of the XML Schema nodes being compared, as well. For example, for INTERIOR-INTERIOR node comparisons, it just uses StructureMatcher, but for LEAF-LEAF comparisons it uses additional information such as SiblingName and data type. The improved results suggest, perhaps somewhat intuitively, that interior nodes are more sensitive to where they appear in the XML hierarchy, but that leaf nodes are more sensitive to their siblings. The LinearComposite matcher, including various optimizations, achieved an average f-measure of 0.901 across all 10 schema pairs (Fig. 6).

Two other optimizations involved re-evaluating neighbor similarities. In the first, MULTIMATCH_OPTIMIZATION, if a schema element had multiple matches, we would re-consider all of its similarities with a slightly less stringent threshold. The other, SIBLING_MULTIMATCH_OPTIMIZATION, was that if a leaf element pair had multiple siblings that were matches, then we would re-compare all siblings of each, primarily with a less stringent threshold. This could catch cases where the leaf nodes were representing the parts of an address, for example. In this case one would expect multiple siblings to match their coun-

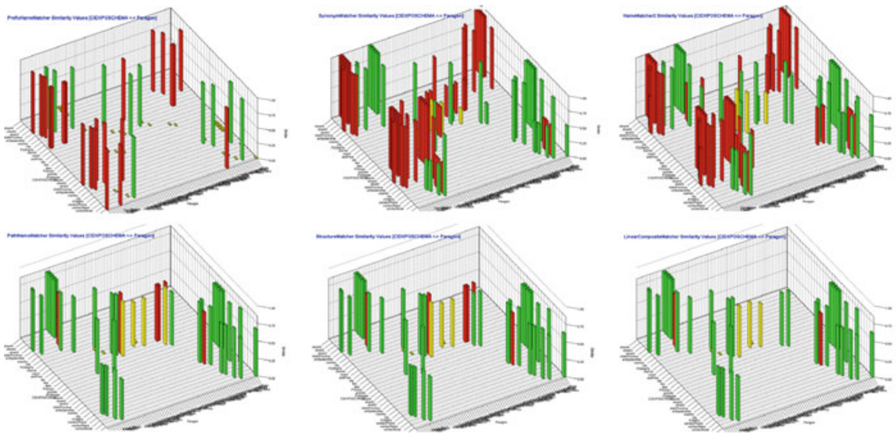


Fig. 6. CIDXPOSCHEMA and Paragon element similarity scatterplots. The matchers and f-measures achieved were (top row, left-to-right) PrefixName 0.268, Synonym 0.548, Name 0.557, (bottom row, left-to-right) PathName 0.854, Structure 0.866, LinearComposite 0.903. Green=true positive, Red=false positive, Yellow=false negative.

Table 2. Effect of turning off some of the individual optimizations on average f-measures for various matchers.

Optimization Turned OFF:	Matcher:					
	PrefixName	Synonym	Name	PathName	Structure	LinComp
NONE	0.433	0.461	0.494	0.820	0.871	0.901
REQUIRE_ELEMENT_NAMES_TO_BE_SIMILAR	0.433	0.461	0.496	0.748	0.715	0.792
MULTIMATCH_OPTIMIZATION	0.449	0.544	0.583	0.831	0.861	0.874
SIBLING_MULTIMATCH_OPTIMIZATION	0.433	0.459	0.492	0.821	0.869	0.898
ELIMINATE_BAD_DATATYPE_MATCHES	0.405	0.456	0.490	0.786	0.868	0.897

terparts in the set of address leaf nodes in the other schema. Finally, there are simple heuristics which can be applied such as eliminating unlikely node combinations with the ELIMINATE_BAD_DATATYPE_MATCHES optimization. For example, ROOT nodes never matched LEAF nodes.

In general, our approach when investigating such optimizations and matcher combination alternatives was to expose the enhancements as parameters so that a human operator could experiment with them via a GUI. As can be seen in the Table 2, individual optimizations can have different effects on matchers, though we highlight those that help most with the higher level combination matchers (Structure and LinearComposite).

5 Related Work

As mentioned in the first section, many methods for creating composite matchers have been tried, and this section explains the difference between them and the proposed approach. One major distinction between these methods is whether they rely on manual tuning of the composition sttructure and parameters, or such parameters are estimated from a training set and verified on an independent test set. The composition methods developed in the COMA [2,6] and GLUE [14] systems are based on manual tuning of the composition parameters, so comparison with learning methods for tuning parameters is not entirely correct; a composite matcher that is manually tuned with a specific set of schemas in mind can certainly be expected to be more accurate than a learning matcher that is tested under a cross-validation protocol.

Among the learning methods for composing matchers, our approach is most similar to the one proposed by Marie and Gal [13], who have approached the problem from a Bayesian network perspective, too, arguing that a disciplined approach to handling match uncertainty has to be applied. However, their approach is based on Naive Bayes networks, that is, two-level Bayesian networks with one root node that corresponds to the matching event, and many leaf nodes that are directly children to the root node. It can be shown that such a Naive Bayes network has the same classification properties as a logistic regression model, and the decision surface is linear, similar to the one used in the LSD and GLUE systems [4,14]. In contrast, a full (non-naive) Bayesian network like the one proposed in this paper can model arbitrary correlations and decision surfaces.

Furthermore, the Bayesian network proposed in this paper is also different from the Bayesian network classifiers used in the YAM system [16] in that our network includes unobservable nodes corresponding to types of matchers; in contrast, YAM employs the BayesNet classifier from the WEKA library that can learn the structure of a fully observable network by adding and removing edges, but cannot add unobservable nodes [17]. Unobservable nodes corresponding to a type of matcher (e.g. lexical, dictionary-based, structural, etc.) present a natural way of representing the conditional dependency between multiple matchers of the same type, because they restrict the edges of the graph only to the nodes of the same type. In contrast, a fully-connected BN without hidden nodes would require an exponential number of CPT parameters to be estimated, which would make it practically impossible to collect the data necessary for estimating them. This problem is further compounded by the continuous values of the similarity values produced by basic matchers — in fact, it is not immediately clear how YAM would have been able to learn a fully connected BN with 13 continuous nodes representing the similarity values of each basic matcher, from the few thousand examples available from the PO dataset under the two LOOCV protocols.

On the other hand, non-linear classifiers such as decision trees [15] can indeed represent non-linear decision surfaces from a limited number of training examples, but are not inherently probabilistic, and the binary decisions output by them are not easy to use in the global assignment process that determines the entire mapping between two schemas from the pair-wise matches between their individual elements. Other probabilistic approaches to the automatic schema matching problem include the use of an attribute dictionary in the AUTOMATCH system, where training examples of matching schemas are used to compile the dictionary, and candidate elements from new schemas are compared probabilistically to the dictionary. Although this approach does result in probabilistic estimates of matches, the compilation of the dictionary requires many training examples, and is best suited to domains where many pairs of schemas have to be matched repeatedly.

6 Conclusions and Future Work

We have proposed a novel method for creating composite matchers for the purpose of automatic schema matching. Its main advantage is the explicit modeling of the conditional statistical dependence between the similarity values computed by individual basic matchers. Experiments suggest that it combines successfully the outputs of such matchers, and achieves matching accuracy significantly exceeding that of the individual matchers. Furthermore, its outputs are estimates of the genuine probabilities of match, which allows the application of decision-theoretic methods for optimal judgment whether elements match, or not. Further work will focus on leveraging the clear semantics of the computed probabilities for improving the accuracy of the global matching algorithm, as well as on improving the computational properties of the proposed Bayesian method.

As a means of comparison and investigation into the limits on the accuracy obtainable by means of matcher composition, we analyzed and identified several typical matching mistakes made by the basic matchers, and devising composition methods that could avoid them, without designing domain-specific matchers. These matchers increased the matching accuracy even further, compared to the Bayesian approach, although it remains to be seen in practice how these matchers would peform in new domains.

References

1. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**, 334–350 (2001)
2. Do, H.H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches. In: Proceedings of the 28th International Conference on Very Large Data Bases (VLDB) (2002)
3. Li, W., Clifton, C.: A tool for identifying attribute correspondences in heterogeneous databases using neural network. J. Data Knowl. Eng. **33**(1), 49–84 (2000)
4. Doan, A., Domingos, P., Halevy, A.: Learning to match the schemas of databases: a multistrategy approach. Mach. Learn. J. **50**, 279–301 (2003)
5. Bergamaschi, S., Castano, S., Vincini, M., Beneventano, D.: Semantic integration of heterogeneous information sources. J. Data Knowl. Eng. **36**(3), 215–249 (2001)
6. Do, H.H., Rahm, R.: Matching large schemas: approaches and evaluation. J. Inf. Syst. **32**(6), 857–885 (2007)
7. Doan, A.H., Domingos, P., Halevy, A.: Reconciling schemas of disparate data sources: A Machine Learning Approach. In: SIGMOD 2001 (2001)

8. Embley, D.W.: Multifaceted Exploitation of Metadata for Attribute Match Discovery in Information Integration. In: WIIW 2001 (2001)
9. Heckerman, D.: A tutorial on learning bayesian networks. J. Learn. Graph. Models, 301–354 (2001)
10. Tang, J., Li, J.Z.: Using bayesian decision for ontology mapping. J. Web Semant. **4**(4), 157 (2006)
11. Thiesson, B.: Accelerated quantification of bayesian networks with incomplete data. In: Proceedings of the Conference on Knowledge Discovery in Data, pp. 306–311 (1995)
12. Pan, R., Peng, Y., Ding, Z.: Belief update in Bayesian networks using uncertain evidence. In: 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), pp. 441–444 (2006)
13. Marie, A., Gal, A.: Managing Uncertainty in Schema Matcher Ensembles. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS (LNAI), vol. 4772, pp. 60–73. Springer, Heidelberg (2007)
14. Doan, A.H., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.: Learning to match ontologies on the semantic web. VLDB J. **12**(4), 303–319 (2003)
15. Duchateau, F., Bellahsene, Z., Coletta, R.: A Flexible Approach for Planning Schema Matching Algorithms. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part I. LNCS, vol. 5331, pp. 249–264. Springer, Heidelberg (2008)
16. Duchateau, F., Coletta, R., Bellahsene, Z., Miller, R.J.: Not yet another matcher. In: Proceedings of CIKM'09, Hong-Kong, China, pp. 2079–2080, November 2009
17. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
18. Berlin, J., Motro, A.: Database schema matching using machine learning with feature selection. CAiSE 2002. LNCS, vol. 2348, pp. 452–466. Springer, Heidelberg (2002)
19. Rajesh, A., Srivatsa, S.K.: XML schema matching – using structural information. Int. J. Comput. Appl. **8**(2), 34–41 (2010)

An SNMP Filesystem in Userspace

Rui Pedro Lopes[1,2(✉)], Tiago Pedrosa[1,2], and Luis Pires[1]

[1] Polytechnic Institute of Bragança, Bragança, Portugal
[2] IEETA, University of Aveiro, Aveiro, Portugal
rlopes@ipb.pt

Abstract. Modern computer networks are constantly increasing in size and complexity. Despite this, data networks are a critical factor for the success of many organizations. Monitoring their health and operation status is fundamental, and usually performed through specific network management architectures, developed and standardized in the last decades. On the other hand, file systems have become one of the best well known paradigms of human-computer interaction, and have been around since early days in the personal computer industry. In this paper we propose a file system interface to network management information, allowing users to open, edit and visualize network and systems operation information.

Keywords: Network management · SNMP · File system

1 Introduction

A file system is a database typically storing large blocks of information. The information is stored in the form of files, structured as a hierarchy of directories. Each entry in the file system, including directories and files, is characterized by a limited group of attributes (instant of creation, instant of the last access, instant of the last change, permissions, owner, group). This paradigm is, perhaps, one of the best well known mechanism for storing information, available in several operating systems as well as in some embedded devices, such as PDAs, mobile phones and even digital cameras.

Usually, files are stored locally in persistent memory, such as hard-disks or memory cards. It is also common, particularly in enterprises, to use network file systems to store files in a remote server, accessed through a special protocol like NFS [1] or SMB. In this situation, the network server exports part of the local file system to a set of selected clients, allowing them to remotely access files and directories. However, this centralized, single server approach, suffers some scalability issues related to throughput, capacity and fault tolerance.

Distributed file systems are designed to improve scalability and fault tolerance by transparently balancing the access between servers. It provides the same view to every client and is responsible for maintaining coherence of data through distributed locking and caching [2–4].

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 124–136, 2013.
DOI: 10.1007/978-3-642-40654-6_8,

Yet another paradigm, cloud storage systems, such as Dropbox[1], transparently synchronize the local copy of data with a remote datacenter, allowing user access to personal and shared files anytime, anywhere.

Based on this paradigm, we considered the possibility of accessing network management information as a set of virtual file systems. Network resources, usually accessed through SNMP [5], COPS [6], or other network management protocol, are seen as remote shares, to be mounted in a regular workstation file system and the instrumentation and configuration information accessed as regular files.

The nature of the information as well as the purpose of this *SNMP File System* (`SNMPFS`) is radically different from traditional distributed file systems (DFS). This makes the current requirements different of traditional DFS. First, distributed file systems are used to store files belonging to one or more users. In a network management scenario, the information is generated both by network resources and users. The former is used for instrumentation and the later is used for configuration.

Second, the content of the majority of network management backed files is constantly changing, because of the dynamic nature of network parameters. As an example, consider a value representing the number of transmitted packets or a value representing CPU load. In regular file systems, used to store personal and application files, entries seldom change. This allows better cache hit rates than in the former situation.

Third, network management files are very small, resulting from the parameters they represent. Frequently, a single `int` is used and, some times, a small table of values is sufficient.

Fourth, the whole `SNMPFS` is composed of several network services and resources, such as routers, firewalls, web servers, and so on. Each resource exports the information resulting from the instrumentation of working parameters thus playing a part in a potentially huge cluster of distributed file systems.

Fifth, faults are usually frequent, resulting from connectivity problems, hardware failures, human action and others. The system should cope with this issues, by recovering when possible and replicating when necessary.

2 File System Design

The goal of the `SNMPFS` is to unify around a unique name space all of the enterprise network management agents. For concept proving, we are using SNMP agents, since they are common in organizations and widely implemented by network devices. This approach allows integrating the tree of management objects of distinct SNMP agents in a single file system. Resuming, the goals of this approach are the following:

[1] http://www.dropbox.com

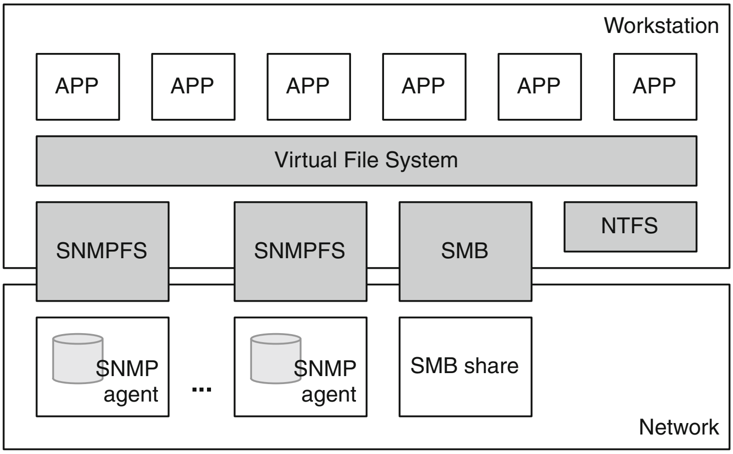


Fig. 1. SNMPFS global architecture.

– allow the use of simple files and directories handling tools (`cd`, `cp`, `cat`, ...);
– allow to consult and change the values through simple tools (file redirection, text editors);
– allow the creation of pipes:
`cat sysUpTime |toxml| mail --s ''sysUpTime'' admin@host.com`;
– allow using office tools, such as Calc or Excel, to view and alter tabular information;
– allow reducing the complexity of the management system.

The information generated by each SNMP agent is accessed through a specific file system (`SNMPFS`), working as a gateway between the regular file operations (`open`, `read`, `write`, `append`, `close`, ...) and SNMP commands (Fig. 1).

As other file systems, like Server Message Block (SMB), also known as Common Internet File System (CIFS), to access SMB (Windows) shares across a network or NTFS for local storage, applications access data through a uniform layer (VFS - Virtual File System). Specific file system details are of the responsibility of each file system technology (SNMPFS, SMB, NTFS and so on). As a result, all the information is stored under the same naming tree.

2.1 Topology

The system topology is straightforward: on one side, several SNMP agents, associated with diverse enterprise resources; on the other side, one or more workstations mount the agents' information through the `SNMPFS` (Fig. 2). It is possible that two or more workstations mount the same agents in its local file system. For read operations this does not present any problem. However, for update operations it is possible that potentially many processes try to change the same value.

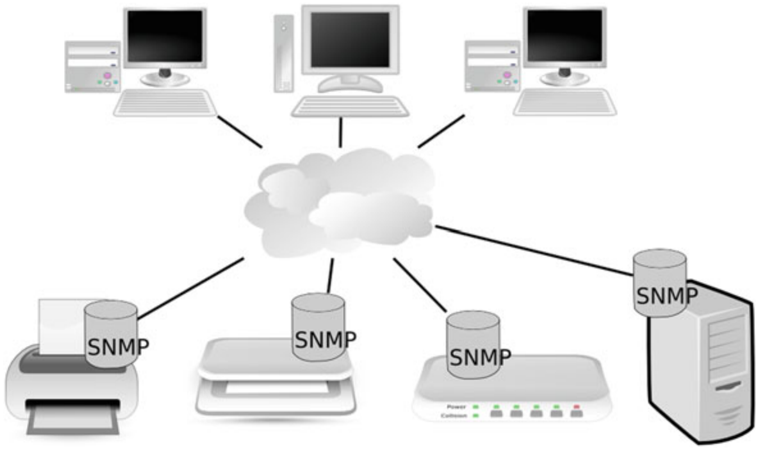


Fig. 2. Simplified topology.

2.2 Locking

In conventional distributed file systems, file locking is essential for coordinating access to shared information among cooperating processes. If multiple processes are writing to the same file it is necessary to regulate the access through some kind of locking mechanism. In `SNMPFS`, locking is performed by the agent, in accordance with the managed object definition, since SNMP agents may also be updated by network management applications concurrently. Some MIBs provide a mechanism to regulate concurrent access. The Expression MIB [7], for example, has tables with a special column used to instantiate the row – `RowStatus` [8].

2.3 Security

SNMP is inherently insecure. Although true for the versions 1 and 2c, SNMPv3 present a modular security architecture based on cryptographic protocols and algorithms. It is mandatory that SNMPv3 implementations support the HMAC-MD5-96 protocol for authentication. They can also support the HMAC-SHA-96 for authentication and the CBC-DES for privacy [9]. More recently, a new privacy protocol was added. [10] describes the Advanced Encryption Standard (AES) for SNMPv3 in the SNMP User-based Security Model which can be used as an alternative to the CBC-DES.

The SNMPv3 security service provides data integrity, data origin authentication, data confidentiality and message timeliness as well as limited replay protection. It is based on the concept of a user, identified by a *userName*, with which security information is associated. In addition to the user name, an authentication key (*authKey*) is shared between the communicating SNMP engines, ensuring authentication and integrity. A privacy key (*privKey*), also symmetric, ensures confidentiality.

Complementing the communication security, the SNMPv3 model also provides access control through a view-based access control model [11]. This model grants or denies access to MIB portions (view subtrees) according to the predefined configuration and the current user permissions.

The security details for SNMPv3, either for authentication, integrity, confidentiality and access control, dictates the security functions for the `SNMPFS`. We have to pass to the file system the authentication and the access control required by the SNMP model. The authentication problem is performed by the system when mounting the file system. A similar approach is followed for NFS or SMB shares:

```
mount -t smb //server/share /mnt -o username=aUser,password=xxx.
```

If the server recognizes the username and password, the host is allowed to access the file system and a user ID (uid) is associated with it.

Access control is enforced by file permissions. In Unix, each file has a set of permissions (read, write, execute) for the file owner, group and others. For example, the permissions

```
-rwxr-x---
```

gives the owner the possibility to read, write and execute the file, the group to read and execute and no other user can read, write or execute.

indent `SNMPFS` translates each file permission to the View-based Access Control mechanism of the SNMPv3.

2.4 Attributes

File system entries, such as files or directories, are characterized by a set of attributes which describes their fundamental aspects, such as size, date, permissions, name and others. The name and number of attributes is typically static, meaning that it is not possible to add or remove further information to each file system entry latter on.

An attribute which is necessary to better describe the data types and the structure of an SNMP agent is the MIB tree it implements. The MIB tree is described in a set of MIB files which contain each node name, data type, restrictions and role. With this information, the `SNMPFS` can present to the user a more meaningful set of file names as well as file types (a table, a string, an int, etc). In particular, this information is valuable for tables, which the `SNMPFS` exports as Coma Separated Values (CSV) format and can be opened and edited by a spreadsheet, such as Microsoft Excel or OpenOffice Calc.

To be able to access the meta-information about management data, the `SNMPFS` has the possibility to load MIB files from a specific directory. This information will allow the files to have a more meaningful name as well as adapting the content to the nature of the information it stores.

3 Implementation Details

The `SNMPFS` implements a gateway between regular file system access primitives and SNMP commands. Generally, file systems are implemented at kernel space, however, we chose to implement the file system in userspace to facilitate the

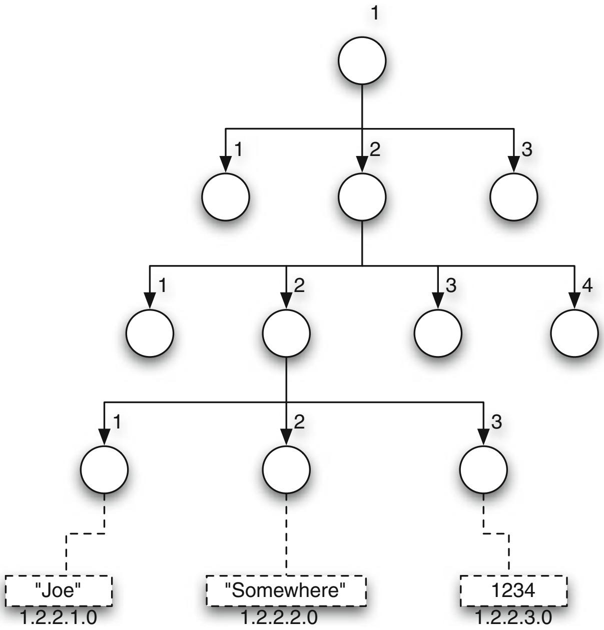


Fig. 3. Scalar values organization in the agent.

development and test process. We used FUSE [12], a stable and well known API for file system development.

The SNMP information is somehow austere, mainly because of the Object IDs (OIDs) that it uses to identify each managed object. The OID is a sequence of integer values, separated by a '.' (dot): 1.3.6.1.2.1.1.1.0. This sequence defines a path in the agent's tree of objects, referring to a specific value (Fig. 3). This path, for example, points to a specific object, which refers to a value resulting from device instrumentation.

In Fig. 3, for example, the OID 1.2.2.1 refers to the lower left node in the tree. This node is associated with a specific value, a scalar, in this case, which contains the string "Joe".The scalar is viewed as an additional node, a leaf, and is referred by adding a '.0' to the OID.

Each object, the circles in the figure, has a set of attributes or meta-information, which allows the user to get the semantics of the value (what does "Joe" stands for). The attributes, as well as the overall structure, is described in a file, called a Management Information Base (MIB), which associates a descriptive, meaningful, name to each object and further describes the data type, access restrictions, OID structure and others. From the user perspective, this also allows mapping the sequence of integers to a short name: it is easier to refer to each object by the short name, instead of the OID (`get sysDescr`, instead of `get 1.3.6.1.2.1.1.1.0`).

After parsing the MIB files, the `SNMPFS` performs this mapping, storing the meta-information to improve the information provided to the user by the file system. The OID in the sequence of integers format will only be used when the MIB is not available.

3.1 The MIB Parser

As mentioned above, the MIB structure is important to SNMPFS to provide a more useful and meaningful view of the file system. MIB files are written in a subset of the Abstract Syntax Notation One (ASN.1), called "Structure of Management Information" [13]. The MIB must be parser so that the tree, nodes, types and objects extracted. We are using Marser [14], an API to parse SMI (v1 and v2).

Moreover, the information from the MIB allows to identify tabular information, which further helps the file system to present the information to the user in a more manageable way. In this case, tables will be available is CSV format, allowing the user to read and modify it using a spreadsheet application.

Tables are represented as a further extension to the OID tree (Fig. 4). As in the previous case, where each scalar is retrieved from a leaf, tabular values are retrieved from several leafs, hanging on the OIDs that represent the columns (in the figure, the table is referred with the OID 1.2.3, which has the columns 1.2.3.1 and 1.2.3.2).

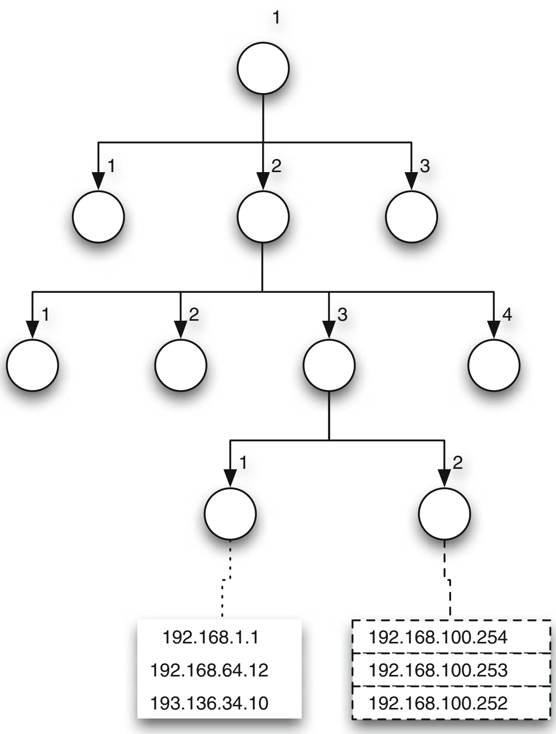


Fig. 4. Tabular values organization in the agent.

One or more of the columns are the index, which allows to retrieve the rows in the table. So, to get the first value of the table, it is necessary to issue: `get 1.2.3.2.192.168.1.1`, which yields `192.168.100.254`.

3.2 Files

When dealing with agents with an unknown structure, the user usually has to explore the management information tree, retrieving all the information that the agent stores. This operation is called 'walk', because it allows to visit all the places "hidden" in the agent. The SNMPFS has a special file which allows doing precisely this. The file, called `_walk.txt`, lists all the managed objects retrieved as the result of a 'walk' operation. By simply opening this file in a text editor, the user will be able to immediately see the objects the SNMP agent implements:

```
sysDescr;1.3.6.1.2.1.1.1.0
sysObjectID;1.3.6.1.2.1.1.2.0
sysUpTime;1.3.6.1.2.1.1.3.0
sysContact;1.3.6.1.2.1.1.4.0
sysName;1.3.6.1.2.1.1.5.0
sysLocation;1.3.6.1.2.1.1.6.0
sysServices;1.3.6.1.2.1.1.7.0
sysORLastChange;1.3.6.1.2.1.1.8.0
sysORID;1.3.6.1.2.1.1.9.1.2.1
sysORID;1.3.6.1.2.1.1.9.1.2.2
sysORID;1.3.6.1.2.1.1.9.1.2.3
sysORID;1.3.6.1.2.1.1.9.1.2.4
sysORID;1.3.6.1.2.1.1.9.1.2.5
sysORID;1.3.6.1.2.1.1.9.1.2.6
...
```

With the information obtained in the file, the user can configure the file system, describing which files should be available and what is the name they should have. The configuration is written is XML and define all the aspects of the file-system: the agent's address, access credentials, MIBs to load, which nodes to show and where to mount:

```
<device name="device">
  <mount dir="tmp" />

  <mibs dir="../mibs/">
    <mib file="SNMPv2-MIB"/>
    <mib file="RFC1213-MIB"/>
    <mib file="IF-MIB"/>
  </mibs>

  <snmp address="192.168.1.1" port="161"
      version="v2c" community="public" />
```

```
  <entries>
    <scalar label="sysUpTime" />
    <scalar label="sysDescr" />
    <table label="ifTable" />
    <table oid=".1.3.6.1.4.1.63.501.3.2.2"
        file="myTable">
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.1" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.2" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.3" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.4" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.5" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.6" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.7" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.8" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.9" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.10" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.11" />
      <col oid=".1.3.6.1.4.1.63.501.3.2.2.1.12" />
    </table>
  </entries>
</device>
```

The previous configuration file will result in the appearance of 5 files in the 'tmp' directory – two tables, two scalars and the `_walk.txt` (Fig. 5).

3.3 Values and Tables

Each file representing a scalar simply has to get the value from the agent each time it is read. In the previous configuration file, the scalar 'sysUpTime' and 'sysDescr' show as files, containing the information from the agent located in 192.168.1.1.

Tables, because of the tree like structure, require more processing. The algorithm we follow is:

```
read configuratiom file;
for each entry
  if is table
    read columns;
    if columns empty
      read columns from MIB;
  for each column
    while has more leafs
      get leaf;
      store in row,column;
end;
```

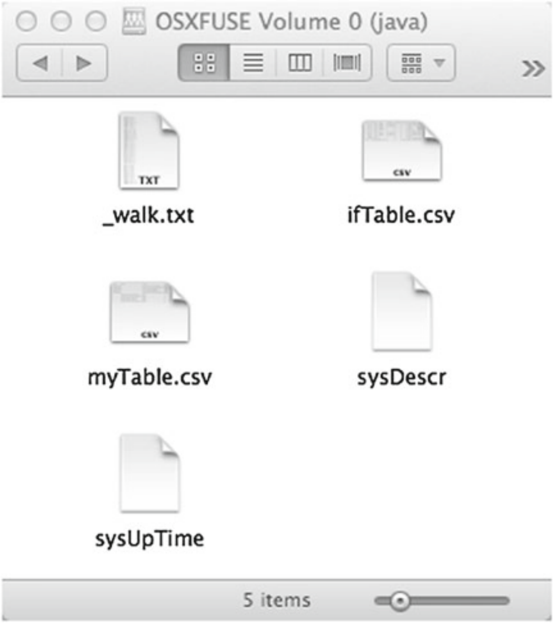


Fig. 5. Screenshot of an SNMPFS directory.

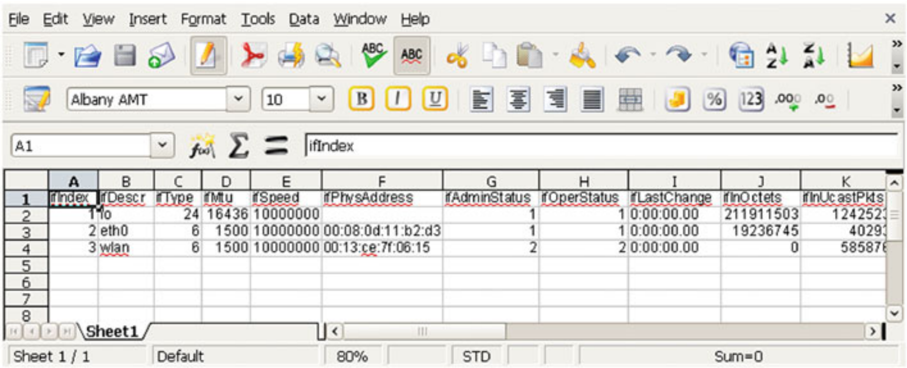


Fig. 6. Editing the ifTable with a spreadsheet.

The fetch of values and store in row and column format allows to build the content around the CSV format, that can be manipulated by a spreadsheet application (Fig. 6).

4 Use Cases

In system and network management, the administrators are used to create scripts to automate some of the typically repetitive and/or boring maintenance activities. Many of this scripts work by reading, writing and updating configuration

Table 1. Monitoring examples.

Query	Object	Type
The number of users on a system	hrSystemNumUsers	Scalar
The bytes transmitted	ifInOctets;ifOutOctets	Table
The storage areas	hrStorageTable	Table
The processor load	hrProcessorLoad	Table

files, altering the way daemons and services work (such as e-mail, HTTP, SSH, ...). The `SNMPFS` enables mapping SNMP information to a file system structure, thus contributing to the integration of monitoring, configuration and accounting processes. In other words, the `SNMPFS` will foster administrators to develop new tools easier and to use the same tools in different scenarios because of the transparency and integration of the file system paradigm.

The extension of the file system with files related to instrumentation and configuration information from network devices will further alleviate the burden. By mounting several devices in the same file system, where each directory represents a different agent, enables the administrators to be able to make queries or change values on all the equipment, just by browsing the file system.

4.1 Monitoring

Monitoring operations require the user to retrieve, process, analyse and visualize instrumentation information. For example, several parameters can be queried to get the status of remote hosts (Table 1).

Other common operation is to build the topology map of the network, representing the connections and hosts structure. This is done with the help of the IP forwarding table, maintained in the switches and routers. Each table gathers the MAC addresses in each port, allowing the correlation of addresses into building a visual representation of the network topology. By replicating this information (a simple copy will do), will enable to create a view of the network a specific times.

4.2 Configuration

One challenge on integrated management of networks is how to apply policies that are transversal to more that one equipment. The integration of several SNMP agents in the same file system can enable the administrator to create sound scripts to apply the policy.

In complement to these use cases, one that is being currently addressed is the use of version control systems for maintaining snapshots of SNMP agents' configuration. Each type and version of equipment has different ways for manipulating their configurations, because of the MIBs they implement. With the `SNMPFS`, administrators can make use of already accepted solutions for version control. A Distributed Version Control, such as GIT[2] enables to have a master repository for each agent as well as a local repository in the management stations.

The first mount of the agent, a new repository is created and the files are added, tagged as the initial commit. This local repository is pushed to the master repository, which will work as another, more general, versioning peer. The master is configured for post commit routines, that will update the directory where the agent is mounted. The master will apply the changes in the file system, configuring the equipment accordingly and changing the properties that enables the updated of the configurations. This enables that different administrators can work on their stations using the repositories for changing configurations and the push the configurations to the master that will change the agents' status. Moreover, using tags to identify specific configuration versions allows to easily change form one configuration to another.

4.3 Scheduling Operations

Modern operating systems have tools that enables users to schedule jobs (commands or shell scripts) to run periodically at certain times or dates. One such tool, popular un Unix-like operating systems, is `cron`, used to automate system maintenance or administration. Proper configured cron jobs allows to activate some options at some time on agents, for example, to shutdown several devices at a specific time. Moreover, it also allows to watch files for specific values for, for example, triggering some configuration change or event (send emails, executing commands).

5 Conclusions

Accessing and updating information is a frequent operation in virtually any activity. Because of the evolution of computing platforms, the electronic information is associated with the concept of files, residing in a generic storage mechanism. Usually, the files are updated by general use applications, such as office suites or drawing editors. Often, the content is in plain text, allowing standard editors to retrieve and update the information.

Because of the ubiquity of files, modern operating system have an extensive set of tools to deal with the maintenance of files, such as renaming, creating, copying, backing up and restore, and so on. In enterprises, work files are typically stored in network storage, made available through a virtual file system in local desktop computers.

Network and system management (NSM) is a major concern for maintaining the system in good working conditions. Many of the tasks involved in NSM require monitoring and updating information resulting from instrumentation procedures in applications, services and equipment. The paradigm in traditional

[2] http://git-scm.com/

NSM models rely on client-server protocols, through special purpose applications.

The SNMPFS, proposed in this paper, integrates network devices, applications and service management in a common platform and paradigm – the file system. In this way, network management operations can benefit from the existing powerful operating system tools to monitor, update instrumentation, configuration and monitoring information.

References

1. Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M., Noveck, D.: NFS version 4 Protocol. RFC 3010 (Proposed Standard), Obsoleted by RFC 3530 (December 2000)
2. Ghemawat, S., Gobioff, H., Leung, S.T.: The google file system. Technical report, Google (2003)
3. Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R.N., West, M.J.: Scale and performance in a distributed file system. ACM Trans. Comput. Syst. **6**(1), 51–81 (1988)
4. Anderson, T.E., Dahlin, M.D., Neefe, J.M., Patterson, D.A., Roselli, D.S., Wang, R.Y.: Servless network file systems. ACM Trans. Comput. Syst. **14**(1), 41–79 (1996)
5. Harrington, D., Presuhn, R., Wijnen, B.: An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. RFC 3411 (Standard) (December 2002)
6. Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R., Sastry, A.: The COPS (Common Open Policy Service) Protocol. RFC 2748 (Proposed Standard), Updated by RFC 4261 (January 2000)
7. Kavasseri, R.: Distributed Management Expression MIB. RFC 2982 (Proposed Standard) (October 2000)
8. McCloghrie, K., Perkins, D., Schoenwaelder, J.: Textual Conventions for SMIv2. RFC 2579 (Standard) (April 1999)
9. Blumenthal, U., Wijnen, B.: User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). RFC 3414 (Standard) (December 2002)
10. Blumenthal, U., Maino, F., McCloghrie, K.: The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model. RFC 3826 (Proposed Standard) (June 2004)
11. Wijnen, B., Presuhn, R., McCloghrie, K.: View-Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). RFC 3415 (Standard) (December 2002)
12. FUSE: Filesystem in userspace. http://fuse.sourceforge.net (2011)
13. McCloghrie, K., Perkins, D., Schoenwaelder, J.: Structure of Management Information Version 2 (SMIv2). RFC 2578 (Standard) (April 1999)
14. Marser: Marser - a mib parser. http://marser.sourceforge.net (2007)

Exploring Similarity Relations According to Different Contexts in Mining Generalized Association Rules

Rodrigo Moura Juvenil Ayres(✉) and Marilde Terezinha Prado Santos

Department of Computer Science, Federal University of Sao Carlos,
Rod. Washington Luis, Sao Carlos, Brazil
{rodrigo_ayres, marilde}@dc.ufscar.br

Abstract. Taxonomies are used in different steps of the mining process, but in the generalization they are mainly used during the pre-processing. In the same way, in fuzzy contexts fuzzy taxonomies are also used, mainly, in the pre-processing step, which is the generation of extended transactions. Besides, it is possible to see that many works have explored mining fuzzy rules and linguistic terms, but few works have explored different steps of the mining process. Moreover, questions like semantic enrichment of the rules have been little explored. In this sense, this work presents *Context* FOntGAR algorithm, which is an algorithm for mining generalized association rules under all levels of fuzzy concept ontologies. Besides, the exploring of rules containing similarity relations according to different contexts will be introduced. In this work the generalization is done during the post-processing step. Other relevant points are the specification of a generalization approach; including a grouping rules treatment, and an efficient way of calculating both support and confidence of generalized rules.

Keywords: Generalized association rules · Fuzzy ontologies · Post-processing · Context-based similarity

1 Introduction

An important task in data mining is the mining association rules, introduced in [1]. In traditional algorithms of association, like Apriori, the rules are generated based only on existing items in the database, and this characteristic makes an excessive amount of rules be produced. In this sense, the domain knowledge, represented via taxonomies, can be used in order to obtain more general patterns, facilitating the user's comprehension. The association task using taxonomic structures is called mining generalized association rules, and was introduced by [2].

In [2], the taxonomy ancestors are inserted into database transactions, called extended transactions, and from such extended transactions it is applied an algorithm for extract the final set of rules, which can be composed of both traditional and generalized rules. However, the inclusion of ancestors in database transactions implies the generation of many candidate itemsets, and ends up generating redundant patterns,

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 137–152, 2013.
DOI: 10.1007/978-3-642-40654-6_9,

making it extremely necessary the use of interest measures for eliminate redundancies. On the other hand, some works, like [2] for example, show that the post-processing stage can be more advantageous, because few candidates and rules can be generated. Moreover, it is eliminated the need of measures used for prune redundant rules, since the process is made based on the traditional patterns generated.

However, in many applications of the real world ontologies and taxonomies may not be crisp, but fuzzy [3], because some applications do not have classes of objects with pertinence criteria precisely defined [3]. In this context, Wei and Chen [3] introduced the use of fuzzy taxonomies. They considered the partial relationships possibly existing in taxonomies, where an item may partially belong to more than one parent. For instance, tomato may partially belong to both fruit and vegetable with different degrees. Wei and Chen thus defined a fuzzy taxonomic structure and considered the extended degrees of support, confidence and interest measures for mining generalized association rules.

However, most of the works are focused into improve methods of to obtain generalized fuzzy association rules, which are the ones composed by linguistic terms, but few works have directed efforts for improve the exploring of generalized rules under fuzzy concept hierarchies, mainly in relation to the stage that they are used.

On the other hand, some works, like [4, 5], explore the semantic enrichment of the rules through similarity relations. these works do not consider that the degree of a similarity relation, between two or more elements, it is also related to the point of view or to the context analysed. For example, consider the problem of compare two vegetables, tomato and khaki, in relation to two different points of view (contexts), appearance and flavour. In respect to the appearance context, would be possible to check that tomato is very similar to khaki, with a very high degree of similarity; but in relation to the flavour, would be possible to check that both are bit similar, with a minor degree of similarity.

Thus, this paper presents the *Context* FOntGAR algorithm for mining generalized association rules, using fuzzy ontologies composed by relationships of specialization/generalization varying in the interval [0,1], and similarity relations with different degrees according to the context. The generalization can to occur in all levels of fuzzy ontologies. The paper is organized as follow: Section 2 shows some related works. Section 3 presents the *Context* FOntGAR algorithm. Section 4 presents the experiments, and the Sect. 5 shows the conclusions.

2 Related Work

Aiming to obtain general knowledge, the generalized association rules, which are rules composed by items contained in any level of a given taxonomy, were introduced by [2]. There are many works using crisp taxonomic structures. These works are distinguished, mainly, in function of the stage (of the algorithm processing) in which these structures are used.

In the pre-processing, the generalized rules are obtained through extended databases, and these bases are generated before the pattern generation. Extended databases are the ones composed by transactions containing items of the original database and ancestors of the taxonomy. In the post-processing the generalized rules are obtained after the generation of the traditional rules, through a sub-algorithm that uses some generalization methodology based on the patterns generated.

In [6], the mining is made using an efficient data structure. The goal is to use the structure for find rules between items in different levels of a taxonomy tree, under the assumption that the original frequent itemsets and association rules were generated in advance. Thus, the generalization occurs during the post-processing step. In relation to the post-processing, [2] proposed the GARPA algorithm. The algorithm, unlike what was proposed by [2], do not insert ancestor items in the database transactions. The generalization was done using a method of replacing rule items into taxonomy ancestors. From the quantitative point of view, this process is more advantageous than proposed by [2], because implies a smaller amount of candidates, and consequently of rules generated, dispensing the use of measures for pruning redundant rules.

In mining generalized rules, most of the works using fuzzy logic are mainly focused into obtain generalized fuzzy association rules, which are the ones composed by fuzzy linguistic terms, such as young, tall, and others. In such approaches are used crisp taxonomies and the linguistic terms are generated based on fuzzy intervals, normally generated through clustering. Besides, these works are directed to explore quantitative or categorical attributes. In this context we can to point, for example, the works [7–11]. On the other hand, few works use fuzzy taxonomies in order to obtain their rules. In this case, the focus is not the exploring of patterns composed by linguistic terms, but it is how to explore taxonomic structures composed by different specialization/generalization degrees.

The problem of mining generalized rules using fuzzy taxonomies was proposed by [3]. They included the possibility of partial relationship in taxonomies, i.e., while in crisp taxonomies the specialization/generalization degrees are 1, in fuzzy structures such degrees vary in the interval [0,1]. So, the degree μ_{xy} which any node y belongs to its ancestor x can be derived based upon the notions of subclass, superclass and inheritance, and may be calculated using the max–min product combination. Specifically,

$$\mu_{xy} = \max_{\forall l : x \to y} \left(\min_{\forall e\, on\, l} \mu_{le} \right) \tag{1}$$

Where l: $x \to y$ is one of the paths of attributes x and y, e on l is one of the edges on access l, μ_{le} is the degree on the edge e on l. If there is no access between x and y, $\mu_{xy} = 0$ [4].

In addition to defining such structures, they also consider extended degrees of support and confidence. The degree of the extended support (*Dsupport*) is calculated based on this μ_{xy}. If a is an attribute value in a certain transaction $t \in T$, T is the transaction set, and x is an attribute in certain itemset X, then, the degree μ_{xa} can be viewed as the one that the transaction *{a}* supports x. Thus, the degree that t supports X may be obtained as follows:

$$\mu_{tX} = support_{tX} = \min_{\forall x \in X} \left((\max_{\forall a \in t} \mu_{xa}) \right) \tag{2}$$

Furthermore, an $\sum count$ operator is used to sum up all degrees that are associated with the transactions in T, in terms of how many transactions in T support X:

$$\sum_{\forall t \in \mathrm{T}} count\,(support_{tX}) = \sum_{\forall t \in \mathrm{T}} count\,(\mu_{tX}) \quad (3)$$

Thus, the support of a generalized association rule $X \to Y$, let $X \cup Y = \mathrm{Z} \subseteq I$, can be obtained as follows, where $|T|$ is the total of transactions in the database:

$$\sum_{\forall t \in \mathrm{T}} count\,(\mu_{tZ})/|\,T\,| \quad (4)$$

Similarly, the confidence ($X \to Y$), called *Dconfidence*, can be obtained as follows:

$$\sum_{\forall t \in T} count\,(\mu_{tZ}) \Big/ \sum_{\forall t \in T} count\,(\mu_{tX}) \quad (5)$$

It is important to say in [3] only the concepts are defined and in [12] the authors proposed two algorithms to realize the mining, one working with the mentioned taxonomies, and other working with these taxonomies and linguistic terms. The first was called FGAR, and the second was called HFGAR, both algorithms use the same concept of extended transactions.

A similar work can be found in [13], however, it is related to the mining generalized quantitative association rules. The authors use two different structures: fuzzy concept hierarchies and generalization hierarchies of fuzzy linguistic terms. In the first, a concept may have partial relationship with several generalized concepts, and the second is a structure in which upper level nodes represent more general fuzzy linguistic terms.

As well as Wei and Chen [3, 13] also use the technique of extended transactions. Besides, it is considered the use of interest measures for prune redundant rules. According to [14], the works using fuzzy taxonomies, like proposed by [3], require the same be static, ignoring the fact they cannot necessarily be kept unchanged. For example, some items may be reclassified from one hierarchy tree to another for more suitable classification.

In this sense, the work [14] introduces an algorithm where the final set of rules generated can be updated according to the evolution of the structures. The evolution can to occur due four basic causes: insertion, deletion, renaming and reclassification of items. Fuzzy taxonomies are used and, as well as [3, 13, 14], the generalized rules are obtained like proposed in [2].

Thus, in respect to the use of fuzzy taxonomies, composed by degrees of specialization/generalization varying in the interval [0,1], the works [3, 13, 14], are the most relevant found in the literature.

On the other hand, some works, like [4, 5] are directed to the semantic of the data mined. They use ontologies for extract associations of similarity existing between items of the database. These relations are represented in the leaves of ontology, but the specialization/generalization degrees are constant 1, like crisp ontologies. The work

[4] is an extension of [5], and the main differences are the introduction of a redundancy treatment and a step of generalizing non-frequent itemsets. However, both algorithms are limited, since generalizes at only one level of ontology (leaf nodes to parents).

As said, these works do not consider the question of context in the similarities represented at the leaves. In this line, the work [15] propose an Upper Fuzzy Ontology With Context Representation (UFOCoRe), an approach that represent multiple relationship strengths in a single ontology, so that it is possible to express different relationship semantics depending on the context chosen. The approach does not define context ontology like the ones used in context-aware systems, but it allows organizing the context information of multiple perspectives in single domain ontology. As described, there are few works dealing with mining generalized association rules under fuzzy taxonomies. Besides, most of the works are inserted in the line of mining generalized fuzzy association rules, which is a concept smoothly different, since for it are used crisp taxonomies and the fuzzy generalized rules are obtained, most of the time, with the utilization of linguistic terms. Besides, it is possible to see a bias, which is the realization of the generalization process exploring fuzzy taxonomies during the pre-processing stage, through extended transactions. In this sense, considering the concept of fuzzy taxonomies, presented in [3], no work to date was proposed for obtain generalized rules during the post-processing stage including the questions of similarity relations considering context.

3 Context FOntGAR Algorithm

The aim of the *Context* FOntGAR is post-process a set of specialized association rules (AR) using fuzzy ontologies, in order to obtain a reduced non-redundant and more expressive set of generalized rules, facilitating the user's comprehension. Figure 1 illustrates all steps of the *Context* FOntGAR algorithm. The steps colored in grey are the main points of our algorithm.

3.1 Main Ideas

The process of generating traditional association rules is based on Apriori [16], and as an mining association rule algorithm, it needs of an user-provided minimum support and minimum confidence parameters to run. Moreover, it needs of a minGen, a side and a context parameters:

- ***minsup***, which indicates the minimum support;
- ***minconf***, represents the mininum confidence;
- ***minGen***, which represents the minimum quantity of descendants in different specialized rules;
- ***minSim,*** which is the minimum similarity used in the reasoner inferences [5];
- ***side***, which represents the side of generalization;
- ***context,*** which represents the context used in the similarity;

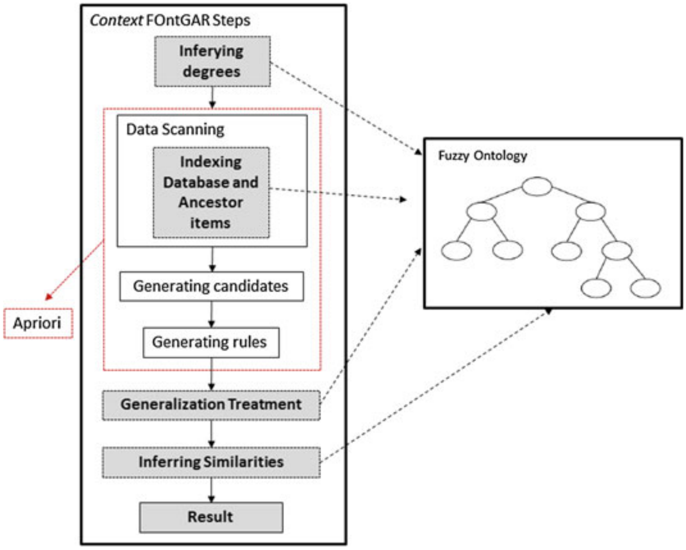


Fig. 1. Context FOntGAR Steps.

The minsup, minconf, minGen and minSim parameters are expressed by a real value in the interval [0,1]. The side parameter is expressed by a string left, right or lr, indicating the generalization side. The generalization can be done on one side of the rule (antecedent or consequent) or both sides (lr: left and right side). While the left side indicates relations between classes of items and specialized items, the side *right* indicates relations between the specialized items and classes of items. The side *lr* indicates relations between classes. The similarities are represented in the leaves of ontology. Relations with similarity degree value greater than or equal to the user-provide minSim [4] can be show in the rules generated, increasing the semantic enrichment of the same. The generalization is made through a sub-algorithm that uses a methodology of grouping and replacement in the rules. In this methodology, two or more rules are grouped in order to be replaced by a unique generalized rule. Several groups can be generated, and the grouping is done based on the parameter side and on the fuzzy ontology. In this case, two or more rules having identical parents in the side of generalization are grouped in a same group.

It is important to say that a group is generated only if two or more rules can be grouped, because is not reasonable generalize a unique rule. As several groups may be generated, various generalized rules may be obtained. During the grouping, the ancestors analyzed are the immediate ones of items present on rules in question, which are the ancestor presents in the current level of generalization. The parameter *side* indicates the generalization side. Thus, when this parameter is set with *left* or *right*, if two or more rules have the same elements in the opposite of *side*, and have identical parents in relation to the items present in the *side*, then these rules are placed in a same group. For example, supposing ontology of bread and milk, where bread is a *breadA, breadB, breadC, breadD, breadE*, and milk is a *milkA, milkB milkC*. Suppose the algorithm generates, during the extracting patterns stage, a set of traditional rules *milkA → breadA, milkA → breadB, milkA → breadC*, which are the ones composed only by leaf nodes.

When the parameter *side* is *lr,* if two or more rules have the same parents in relation to the antecedent items, and, respectively, have the same parents in relation to the consequent items, then these rules will be grouped together. For example, considering that traditional rules *milkA → breadA, milkB → breadB, milkC → breadC* have been generated. Comparing these rules, we can see that they have the same parent in relation to the antecedent, and respectively, they have the same parent in relation to the consequent. Thus, these rules will be grouped together.

It is important to say the rules used in the grouping can be composed by any quantity of items. At first, the patterns used during the generalization are the traditional ones generated by the extracting patterns stage. Posteriorly, the obtained generalized rules are treated in the same way, in order to obtain a new set of generalized rules. Thus, it is a recursive process. An important point is that generalized rules can be generated without the use of all descendants of an ancestor. In this sense, to avoid an over-generalization, a set of specialized rules contained in a group can be substituted by a more general rule only if a minGen parameter [4] was satisfied. Consider that the minGen value is 0.6 (60 %), and the side is lr, the rule milk → bread will be generated even if there is no rule for each kind of bread and milk in the current group, but only if 60 % of descendants of bread and milk are present in this set of rules. Thus, the use of minGen could produce a semantic loss. In this sense, in order to guide the user's comprehension, the algorithm show the items which have not participate in the generalization process. For example, suppose the item breadE is not present in the specialized AR set, the generalized rule are shown as milk → bread (-breadE), indicating that the item breadE did not compose the generalization.

In this research, for represent a fuzzy ontology with specialization/generalization degrees varying in [0,1] and context in similarity relations, we follow the ideas described in two meta-ontologies, proposed in [15, 17] respectively. Both are upper ontologies as it represent fuzzy constructs to be inherited and/or instantiated by specific domain ontologies. Such ontologies are based on OWL DL [17], a W3C recommendation supported by several reasoners and application programming interfaces used to develop ontology-based applications.

3.2 Step by Step

First, the ontology reasoner is used to infer the membership degrees of the leaves in relation to the ancestors, through the Eq. 1 of the Sect. 2. These degrees are stored in a data structure. The steps of data scanning, generating candidates and generating rules are done similarly to the Apriori.

At end of generating rules we have a set of specialized rules, which will be used on the generalization treatment. Then, the generated rules and the side of generalization are passed to the groupingRules function (line 7), which is responsible by the grouping treatment mentioned above. Posteriorly, for each group generated, all rules in a group are represented by a more general rule (line 10). So, the minGen parameter (line 11) is checked, besides, it is verified if antecedent ∩ consequent = 0 and if no consequent item is ancestor of any antecedent item (line 12). If such verifications are satisfied (line 13), the calculus of support is done. If the general rule is not frequent then the generalization is not made. In this case, if the level is 1 (line 19), the rules of the corresponding group are inserted in the result. But if the general rule is frequent, the rules of the corresponding group are replaced by the same, and it is inserted in the result.

After that, if there are generalized rules, the same are used in the next level of generalization. If this situation is true for all next levels, the generalization process will be done until a level below the ontology root. However, if there is no generalized rule at a certain level, then will be impossible generalize in the next levels. When this happens, the generalization process is concluded. After the generalization treatment, the algorithm uses the ontology reasoner to obtain the similarity relations. So, these relations are used in the non-generalized rules. Finally, after that, the algorithm enters its final stage, which is the results generation.

3.3 Calculating the Support and Confidence Values

Considering the fuzzy taxonomy of Fig. 3, *Fruit* → *Meat* is a generalized rule and {Fruit, Meat} is their itemset format. The support is calculated based on the sum of all degrees of transactions that support simultaneous occurrences of {Fruit, Meat}. However, {Fruit, Meat} is obtained and known only during the post-processing. Then, for obtain the degree of each transaction, it would be necessary a new scanning in the

Pseudo-Code of the Generalization Treatment

```
1 if (side = left) or (side = right) or (side = lr) then;
2      level:= 1;
3      nonGeneralizedRules:= all traditional rules;
4      while (level < total of levels) do
5        ontologyVerification(nonGeneralizedRules);
6        aux:= result of ontologyVerification;
7        groupingRules(result of nonGeneralizedRules, aux and side);
8        groupedrules:= result of groupingRules;
9        for (all groups in groupedRules) do
10           all rules in a group are represented by a general rule;
11           verify if the minGen is satisfied;
12           verify other generalization criteria;
13           if (replacement can occur) then
14             do the calculus of support of the general rule;
15           end if
16           if (the general rule is frequent)
17             all rules of the group are replaced by the general rule;
18             generalizedRules:= the general rule;
19           else if (level = 1) then
20               rules of the group will be show in the result;
21       end for
22       if (generalizedRules contains generalized rule) then
23           level: level + 1;
24           for (all rules of generalizedRules) do
25             add the rule generalized into nonGeneralizedRules;
26           end for;
27       end if;
28       if (generalizedRules is empty) then
29         break;
30     end while;
31 end if;
```

Fig. 2. Generalization (Pseudo-Code).

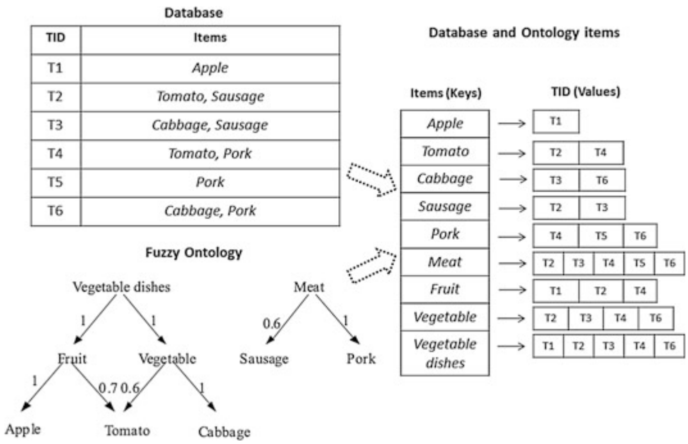


Fig. 3. Indexing items and ancestors.

database. As many generalized rules may be generated, the quantity of new scanning also may be huge, and depending on the quantity of rows of the database, the performance of the algorithm would be affected.

In *Context* FOntGAR we use two data structures (Figs. 3 and 4) to allow the calculating of support avoiding additional scan. Such structures are composed by keys and values. In Fig. 3, a key is an item of the database or an ontology ancestor. Each key points a value, which is a vector storing the transaction identifiers where the key appear. The vector is an object of the class Vector in Java, dynamically created. The equation used in the calculus of support is derived of the Eq. 2 (Sect. 2). So, if we partitioned the same in two subparts (Part 1 and Part 2), we have:

- **Part 1** $= \text{Max}_{\forall a \in t} \, (\mu_{xa})$
- Part 2 $= \text{Min}_{\forall x \in t} \, (\textbf{\textit{Part 1}})$.

As said, we can have many generalized rules, but we don't know what will be generated. So, the itemset format of each may be any $X = \{x_1,\ldots,x_n\}$, where X is the generalized rule, and $x_1,\ldots,x_n$ are items of the rule. That way, during the first scan, we do the computation of Part 1, which is the degree that each transaction t supports an ancestor x. Based on the results of Eq. 1, found at beginning of the algorithm, these degrees are calculated and stored in a data structure (Fig. 4), where a key is the ancestor x (which will be present in generalized rules), and each key points a value, which is a vector storing the degrees mentioned. Thus, since the result of Part 2 correspond to min operator for the degrees related to any rule $\{x_1,\ldots,x_n\}$, we use the stored degrees of $x_1,\ldots,x_n$ for calculating the Part 2, obtaining the support of any generalized rule.

An important point is that if $\mu t_X = 0$ the transaction does not supports x_n, then the degree $\mu t_X = 0$ is not stored in the vector. Thus, each vector linked in a key of the Fig. 3 has the same quantity of positions of the vector pointed out by the same key of the Fig. 4. Besides, in such vectors, the values of correspondent positions are related.

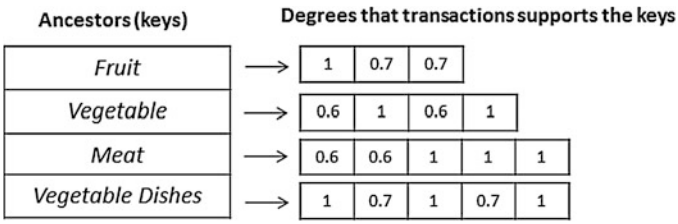


Fig. 4. Storing the transactions support degrees.

For example, through Fig. 3 we can see that the key Fruit is present in three transactions, T1, T2 and T4. Then, from the Fig. 5 we can infer that the degree which T1, T2 and T3 support Fruit is 1, 0.7 and 0.7, in the same order.

Now, consider an example about how calculate the support of the rule *Fruit* → *Meat*: First, the algorithm uses the structure shown in the Fig. 3 for verify the quantity of transactions in the intersection of values stored in vectors of these keys, since it represents all simultaneous occurrences of Fruit and Meat on the dataset transactions. Figure 5 illustrates this idea. In this case we have two occurrences of {*Fruit, Meat*}.

Then, in relation to each key, the algorithm uses the positions of these transactions in Fig. 3 to found the degree which each transaction supports these ancestors. Such degrees are present in the same positions of the vectors linked at Fruit and Meat on the Fig. 4. In this case we have: Fruit: 0.7/T2, 0.7/T4; Meat: 0.6/T2, 1/T4, which are results of Part 1. Based on these degrees, we use Part 2 to calculate the μ_{tX}, where X is {*Fruit, Meat*}.

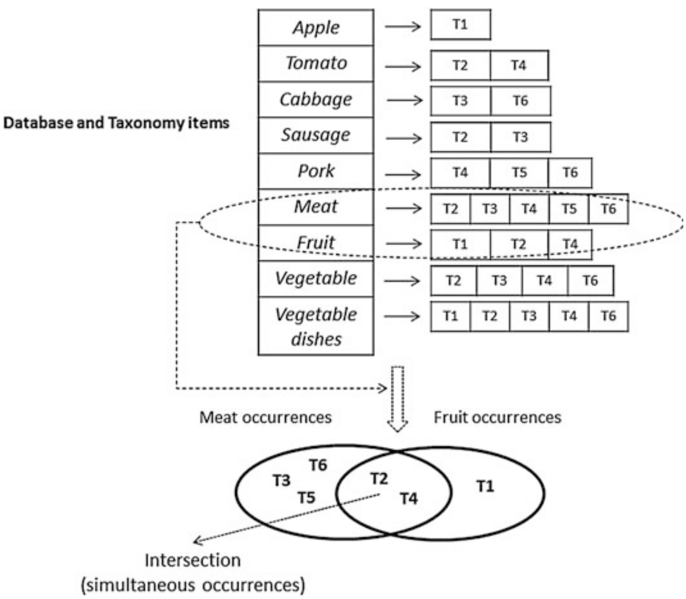


Fig. 5. Idea Used in the calculating support.

For T2 we have:

$$\mu_{tX} = \min_{\forall t \in X} (Part\,1) = \min(0.7, 0.6)\,0.6$$

For T4 we have:

$$\mu_{tX} = \min_{\forall t \in X} (Part\,1) = \min(0.7,\ 1)\,0.7$$

So, according to Eq. 3, we have 0.6 + 0.7 = 1.3. Furthermore, the Eq. 4 is used to calculate the support, which is 0.21. Although we presented a specific example, the process applies to any rule.

3.4 Obtaining Similarity Relations According to Different Contexts

As said before, for represent our fuzzy ontology, we follow the ideas described in two meta-ontologies, proposed in [15, 17]. The approach proposed in [15] allows to represent, in a single ontology, distinct relationships according to different contexts.

In relation to fuzzy relationships, they introduce the *ctx:ContextFuzzyRelationMembership* class, responsible for associating fuzzy relationships to several contexts. *Ctx:ContextFuzzyRelationMembership* is subclass of the *fuzz:FuzzyRelationMembership* class from the fuzzy ontology, thus it inherits *fuz:fuzzyRelationDomain*, *fuz:fuzzyRelationRange*, *fuz:fuzzyRelationProp* and *fuz:membershipDegree* properties. The context association is represented by *ctx:hasContext* and *ctx:context* properties, which link contexts to fuzzy relationships (*fuz:FuzzyRelation*) and fuzzy degrees respectively. By using such constructs, a domain expert can model fuzzy relationships from different perspectives, with specific fuzzy degrees according to each context.

In our algorithm, the similarity degree values between items are represented in the fuzzy ontology leaves, which specify the semantics of the database contents. This step navigates through the fuzzy ontology structure to identify semantic similarity between items, according to the pre-defined *context* parameter. If according to a user-provide *context* the similarity degree between items is greater than or equal to the minSim parameter cited in Sect. 3.1, a semantic similarity association is found and this association is considered similar enough. A fuzzy association of size 2 is made by these pair of items found and are expressed by the symbol $\sim$ indicates the similarity relation between items, for example, $item_a \sim item_b$.

After that, this step verifies the presence of similarity cycles as proposed in [18]. These are fuzzy associations of size greater than 2 that only exists if the items are, in pairs, sufficiently similar. The minimum size of a cycle is 3, and the maximum is the number of sibling leaf nodes, for example, $item_a \sim item_b \sim item_c$. According to [18], based on the concept of fuzzy intersection, the similarity degree value of a cycle is the minimum value found among the pairs. For example, if in a context $item_a \sim item_b$ are 0.8 similar; $item_b \sim item_c$ are 0.7 similar; $item_a \sim item_c$ are 0.5 similar, then $item_a \sim item_b \sim item_c$ are 0.5 similar. Similarity cycles are obtained through the transitive property [3]. All similarity relations and similarity cycles with degree values greater than or equal to the *minsim* are stored (as strings) by the algorithm. After that, this step does a search in the rules generated checking if the same have items that are included in some relation or cycle stored. In positive cases, these items are replaced by the correspondent string stored. We can say the positive cases are related to the traditional rules which have not been generalized, since the similarity relations are associated only to the leaf nodes. For example, suppose the rule: $item_a$, $item_d$ $item_b \rightarrow item_f$, $item_h$. Considering that there is a similarity relation $item_a \sim item_b$, then the stored correspondent string, $item_a \sim item_b$, it is inserted in the rule, replacing the single items $item_a$ and $item_b$. So only the new rule, $item_a \sim item_b$, $item_d \rightarrow item_f$, $item_h$, it is show by the algorithm.

We can say that our approach is totally different than [4, 5]. In these works, the inclusion of similarities in the rules is done through a concept of fuzzy item, which are a type of similarity representation. Such items are inserted in the set of candidates, during the candidate generation, and are used to generate the rules. Besides, a calculus of fuzzy occurrences also is done. Another different point is that [4, 5] do not consider the inclusion of context in the similarity relation.

4 Experiments

This section shows some experiments performed to validate the *Context* FOntGAR algorithm. Two real datasets were used. The first dataset (DB-1) contains information about Years of study, Race or ethnicity and Sex, and was provided by Brazilian Institute of Geography and Statistics (IBGE). DB-1 contains 10000 transactions with 12 distinct items. The second data set (DB-2) contains a one day sale of a supermarket located in São Carlos city. DB-2 contains 1716 transaction with 1936 distinct items.

Two fuzzy ontologies were created, one for the DB-1, called Ont-1 ontology, and other for the DB-2, called Ont-2 ontology. The Ont-1 was constructed contained one level of abstraction, except by the root, and Ont-2 was constructed with four levels of abstraction, except by the root. In both ontologies the average value of specialization/generalization degrees was 0.8. Both ontologies were modeled in OWL (Web Ontology Language) and the Jena Framework was used to allow navigation through ontology concepts and relations.

In order to compare and illustrate the performance of *Context* FOntGAR, the experiments were carried out. With the DB-1, the GARPA algorithm [2] under a corresponding crisp taxonomy, NARFO [4] under a corresponding crisp ontology and *Context* FOntGAR algorithm under the Ont-1 were run. The purpose was to show what the effect of fuzzy extensions could be. In this comparison, 2 experiments have been conducted.

4.1 Performance Comparison

We performed 2 experiments with real data and taxonomic structures mentioned above, changing a different parameter in each experiment. The experiments were done with default values of parameter, except for the one being varied. By default, minsup = 0.02, minconf = 0.4 and mingen = 0.2. The side of generalization was set to lr in all algorithms.

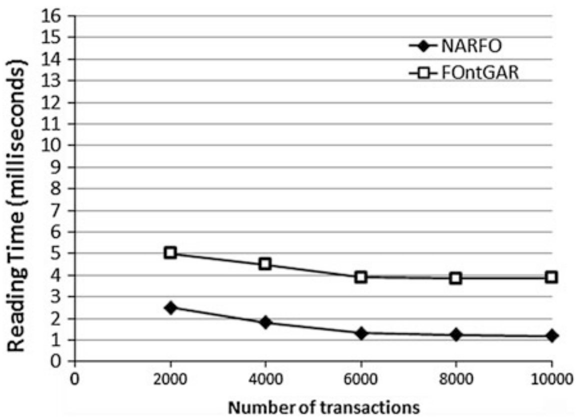


Fig. 6. Scanning time per transaction.

Number of Transactions. In Fig. 6, the vertical axis is the average of reading time per transaction (in milliseconds) in relation to the first scanning in the database. Here was compared the first scan on NARFO and the first scan on *Context* FOntGAR. We varied the number of transactions from 2000 to 10000. From Fig. 6, it is possible see that the gap between *Context* FOntGAR, and NARFO show that the scanning with fuzzy ontologies is more time consuming than scanning with crisp ontologies. There are two reasons. First, the membership degree calculation demands more time. Second, the data structures generation contributes for increase the runtime. However, we can see that the gap tends keep stable with the increase of the number of transactions. This shows that the computational complexity is linear with the number of transactions, which is the same as the crisp algorithm. The difference between the two curves turns to be constant.

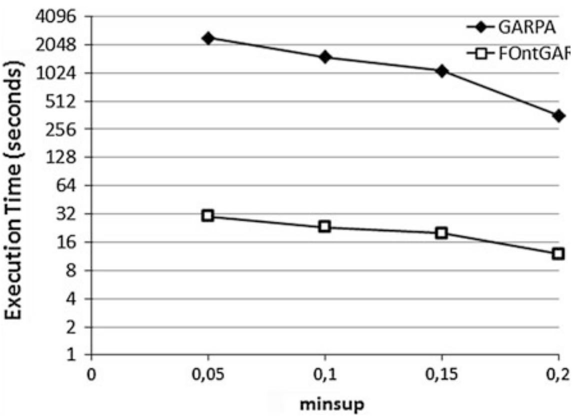


Fig. 7. Comparison in relation to the runtime.

Minimum Support Degree. In Fig. 7 we changed the minimum degree of support from 0.05 to 0.2 %. The vertical axis is the total execution time in seconds. Notably, with the increase of minsup, the runtime of both *Context* FOntGAR and GARPA decreases. The reason is that when the minsup increases the amount of traditional rules decrease, and consequently a minor quantity of rules are post-processed. However, we can see that GARPA consumes more time than *Context* FOntGAR. The reason is that GARPA demands more time during the calculating of support, because a new scan is done in the database for each generalized rule obtained. So, depending on the quantity of rules and rows of the dataset, the runtime can be very high. On the other hand, apart from provide an indexed access to data, in *Context* FOntGAR, the data structures avoid the necessity of new scans in the database, decreasing the runtime.

4.2 Exploring Similarity Relations

In order to explore rules with similarity relations the DB-2 and Ont-2 were used. For explore different contexts Ont-2 was extended through the meta- ontology mentioned above. Two contexts were inserted, flavor and appearance. Table 1 shows some leaf items and their respective similarity degree values, in relation to the two contexts. The part shown represents the similarTo relationship between the spinach and mustard according to context appearance. The similarity degree is set to 0.7.

In Table 1 the similarity degree values are given in pairs of items. For example, spinach and mustard have similarity 0.7 in context of appearance. Besides, based on the Table 1 two similarity cycles can be found in the ontology. Depending on the similarity value, the selection of context may cause change in the similarities represented in the rules. Our experiment was carried out employing the parameters values: minimum support (minsup) = 0.2, minimum confidence (minconf) = 0.2, and minimum similarity (minsim) = 0.3. Some examples of rules generated are:

Appearance Context:

- spinach ~ lettuce ~ mustard, coffee → onion, potato
- tomato ~ khaki, bread → soap, detergent

milk → EuropeanChocolate ~ BrazilianChocolate

Table 1. Similarity degree values

Similarity contexts			
Items		Appearance	Flavour
Coca-Cola	Pepsi	0.8	0.6
Pepsi	Brazilian Coke	0.8	0.5
Tomato	Khaki	0.7	0.3
European Chocolate	Brazilian Chocolate	0.8	0.6
Spinach	Lettuce	0.7	0.4
Spinach	Mustard	0.7	0.4

5 Conclusions

This paper proposes the *Context* FOntGAR algorithm, a new algorithm for mining generalized association rules under all levels of fuzzy ontologies, including similarity relations in the rules. The experiments show that *Context* FOntGAR makes an efficient generalization treatment. This work presents several contributions. First, it is shown an algorithm which uses fuzzy ontologies with context-based similarity relations during the post-processing stage. Considering the bias found in the literature, our algorithm makes an important improvement on the state of the art. Another important contribution is that *Context* FOntGAR improves the semantic in the rules and generates non-redundant patterns without use pruning measures, since the generalized ones are obtained based on the traditional rules. For future works we are doing some improvements in the *Context* FOntGAR algorithm. We are improving the use of *mingen*, based on the user's preferences.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases, Washington, DC, USA, pp. 207–216 (1993)
2. Srikant, R., Agrawal, R.: Mining generalized association rules. Presented at the Proceedings of the 21th International Conference on Very Large Data Bases (1995)
3. Wei, Q., Chen, G.: Mining generalized association rules with fuzzy taxonomic structures. In: 18th International Conference of the North American, Fuzzy Information Processing Society, NAFIPS, pp. 477–481 (1999)
4. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
5. Miani, R.G., Yaguinuma, C.A., Santos, M.T.P., Biajiz, M.: NARFO Algorithm: Mining Non-redundant and Generalized Association Rules Based on Fuzzy Ontologies. In: Filipe, J., Cordeiro, J. (eds.) ICEIS 2009. LNBIP, vol. 24, pp. 415–426. Springer, Heidelberg (2009)
6. Escovar, E.L.G., Yaguinuma, C.A., Biajiz, M.: Using fuzzy ontologies to extend semantically similar data mining. In: Presented at the 21 Brazilian Symposium on Databases, Florianópolis, Brazil (2006)
7. Wu, C.-M., Huang, Y.-F.: Generalized association rule mining using an efficient data structure. Expert Syst. Appl. **38**, 7277–7290 (2011)
8. Chiu, H.-P., Tang, Y.-T., Hsieh, K.-L.: A cluster-based method for mining generalized fuzzy association rules. In: First International Conference on Innovative Computing, Information and Control. ICICIC '06, pp. 519–522 (2006)
9. Mahmoudi, E.V., Sabetnia, E., Torshiz, M.N., Jalali, M., Tabrizi, G.T.: Multi-level fuzzy association rules mining via determining minimum supports and membership functions. In: Second International Conference on Intelligent Systems, Modelling and Simulation (ISMS 2011), pp. 55–61 (2011)
10. Cai, C.H., Ada Fu, W.C., Cheng, C.H., Kwong, W.W.: Mining association rules with weighted items. In: International Database Engineering and Application Symposium, pp. 68–77 (1998)
11. Hong, T.P., Lin, K.Y., Wang, S.L.: Fuzzy data mining for interesting generalized association rules. Fuzzy Sets Syst. **138**, 255–269 (2003)

12. Lee, Y.-C., Hong, T.-P., Wang, T.-C.: Multi-level fuzzy mining with multiple minimum supports. Expert Syst. Appl. Int. J. **34**, 459–468 (2008)
13. Chen, G., Wei, Q.: Fuzzy association rules and the extended mining algorithms. Inf. Sci. Inf. Comput. Sci. Int. J. **147**, 201–228 (2002)
14. Keon-Myung, L.: Mining generalized fuzzy quantitative association rules with fuzzy generalization hierarchies. In: IFSA World Congress and 20th NAFIPS International Conference, pp. 2977–2982 (2001)
15. Wen-Yang, L., Ming-Cheng, T., Ja-Hwung, S.: Updating generalized association rules with evolving fuzzy taxonomies. In: IEEE International Conference on Fuzzy Systems (FUZZ) (2010)
16. Cerri, M.J., Yaguinuma, C., Ribeiro, M.X., Santos, M.T.P.: UFOCoRe: exploring fuzzy relations according to specifics contexts. In: Presented at the International Conference on Software Engineering and Knowledge Engineering (SEKE 2010), San Francisco Bay, USA (2010)
17. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Presented at the Conference on Very Large Databases (VLDB), Santiago, Chile (1994)
18. Smith, M.K., Welt, C., McGuinness, D.L.: W3C Proposed Recomendation: OWL Web Ontology Language Guide. (2004). Accessed 2 Dec 2004

Part III

Artificial Intelligence and Decision Support Systems

Models for Human Computer Interaction in Scheduling Applications

Anna Prenzel(✉) and Georg Ringwelski

Department of Electrical Engineering and Computer Science,
Hochschule Zittau/Görlitz, Brückenstraße 1, 02826, Görlitz, Germany
{aprenzel, gringwelski}@hszg.de

Abstract. There are many algorithms to solve scheduling problems, but in practice the knowledge of human experts almost always needs to be involved to get satisfiable solutions. However, human computer interaction in scheduling applications is often designed in a way, that does not leave much room for own decisions to the user. In this paper, we describe a set of decision support features that can be used to improve the human-computer-interaction in scheduling applications. Based on a study with 35 test subjects and overall 105 h of usability testing we verify that the use of the features improves both quality and practicability of the produced schedules.

Keywords: Human factors · Planning and scheduling · Decision support system · Automation · Interactive scheduling

1 Introduction

Scheduling solutions to support human decisions are widely asked for in several application domains. Very often these solutions turn out in practice to work as sociotechnical or mixed initiative systems. Numerous (human) agents and stakeholders as well as software systems are involved in decision making [8].

Problem Description. In this paper we focus practical scheduling problems, which are products that are sold to several companies. They have similar, but never uniform problems and workflows. The customers require interactive scheduling features including:

- adapting schedules during execution due to accidents that must be resolved immediately
- adapting future schedules due to expert knowledge which was not included in the model a priori.

Another problem is the acceptance of the product by end-users [19]. In interviews with human schedulers we have observed that

- they fear that a system could replace their work and are reluctant to accept push-the-button-optimizers

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 155–171, 2013.
DOI: 10.1007/978-3-642-40654-6_10,

- it is inevitable that expert knowledge on the scheduling process is maintained in a company.

From this point of view we must find appropriate ways to incorporate human factors in the computer-supported scheduling process.

Contribution. In order to target these requirements we define several human–computer interaction models based on an analysis of human decision-making. They can be distinguished by their level of automation that varies between manual and fully automatic.

- We show that human operators should be able to choose the level of automation for each scheduling problem individually.
- We compare the models based on an empirical study we carried out in 105 h of usability testing with 35 test subjects. Our study shows that the quality of the produced schedules correlates with use and availability of (semi-)automatic optimization features.

2 Investigating the Human Contribution to Scheduling

2.1 The Common Structure of Scheduling Problems

The main concern of scheduling is the assignment of *jobs* to *resources*. Jobs are services that must be carried out by the resources, for example, items for production, items for transport or shifts in a hospital. Machines, vehicles or employees can be considered as resources. Scheduling systems are expected to solve combinatorial problems such as finding sequences or start times of jobs, good resource utilization, minimal makespan and many more. Solving these problems is complex (often NP-complete) because solutions have to satisfy numerous constraints including

Start Time Constraints: *For individual jobs, such as* "each job has a time window that restricts earliest and latest possible start time".
Among several jobs, such as "jobs must not overlap in time if they are assigned to the same resource".

Resource Constraints: *For individual jobs, such as* "each job has a set of resources it can be assigned to".
Among several jobs, such as "a limited set of resources can be used at a time".

In addition to meeting the constraints the goal of scheduling is to achieve a good quality of the schedules. It is expressed in terms of one or more objective functions. For example:

– Minimize the total travel time between each two jobs in the schedule (cost function).
– Minimize the time between the beginning of the first and the end of the last job in the schedule (execution time, makespan).

The calculation of the objective functions is application-specific. As an example, a complete formal description of a vehicle routing problem including constraints and objective functions can be found in [3].

2.2 Preferences and Modifications

We have gathered information about scheduling issues in several projects with domain experts in scheduling. Each company has its specific technical requirements on their schedules. Scheduling applications usually consider many, but not all of these requirements [11, 16, 21]. Consequently, they have to be incorporated by the dispatcher. We examine his activity with regard to *preferences* and *modifications.*

Preferences. The dispatchers know the criteria that make their schedules practicable or impracticable and prefer certain schedules over others [12]. Their preferences arise dynamically from their experience and from their expert knowledge [2]. Dispatchers do not only care about the optimization of cost functions, but also

- try to anticipate future events and prepare for them with robust schedules, that can be easily adapted [21]
- try to fulfill individual scheduling wishes of the staff or of customers [12]
- try to avoid problems with the execution of schedules that are known from the past [21].

In order to incorporate his knowledge the dispatcher has to make decisions and incorporates them in a currently valid schedule, which affects the result of optimization goals. Consider the following types of preferences:

Start Time Preferences: "start this job not until 10 o'clock"; "start this job as early as possible".

Resource Preferences: "use resource X (not) for this job"; "use only half of the jobs for this resource".

Optimization Preferences: "reduce the travel time for this resource"; "reduce the overall execution time"; "change the weight of this objective function".

Preferences are not additional constraints, but they represent desired local or global properties of a schedule. However, the human expert often is uncertain about his preferences [15, 20]. In many cases it is not clear from the start whether and to what extent preferences can be implemented. It depends on the impact they have on the overall schedule and particularly on how much the remaining jobs are changed. The human expert decides, if the consequences resulting from his preference can be accepted, otherwise he withdraws or modifies it. He usually compares and evaluates different scheduling scenarios until he decides for one schedule.

Modifications. Subsequent modifications of schedules play a big role in practical scheduling. For different reasons *unanticipated* changes to schedules are frequently. For example, a schedule has to be adapted if a resource breaks down or a new job has to be included in case of an event. Modifications can be considered as changes to the

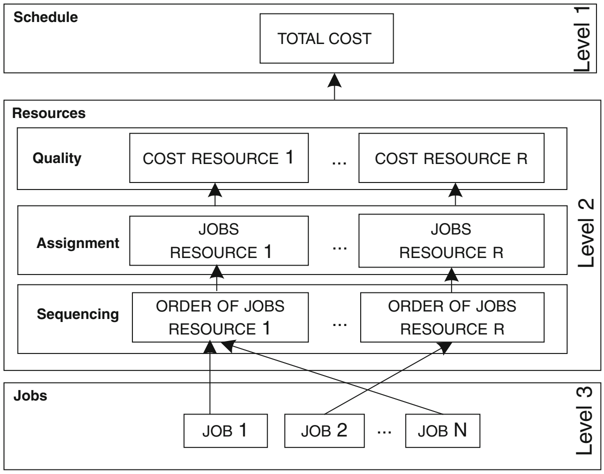


Fig. 1. Abstraction levels for scheduling tasks.

constraints, variables or domains of the underlying formal model to make it reflect a new situation or changed technical requirements. Again, there might be *preferences* about the best way to perform modifications.

2.3 Abstraction Levels of Scheduling Actions

Human operators tend to have an intuition about how to adapt a schedule such that a preference is considered. They use mental models containing as much details of the system as needed to plan the scheduling actions that lead to the desired state of the schedule [5, 9]. The possible levels of detail a schedule provides can be represented in an abstraction hierarchy.

The hierarchy we choose is shown in Fig. 1. From top to bottom, it reveals different levels of detail of a general schedule. At level 1 the only information used are the objective values of the overall schedule. The values of different objective functions can be compared in order to assess the quality of the schedule. The underlying level 2 reveals details of the sub-schedules for each resource. At the "quality"-sublevel there are the same objective functions as in level 1, but with regard to the subschedule of a particular resource. The levels below include the assignment of jobs to resources and, zooming in further, the order of the particular jobs. The lowest level 3 contains the individual jobs. They hold their current assignment (start times, resource affiliations) as well as their constraints for assignments (time windows, valid resources).

A scheduling action at a certain level can be defined without information of the underlying levels. Consider for instance the goal of changing the resource affiliation of a job. It will be irrelevant for the human operator where the job is positioned within the sequence of jobs or at which time it starts. However, for the preference to take effect a decision about the start time has to be made in order to obtain a schedule that does not violate constraints. That means, the level a preference targets and the level at which it is implemented can be different. We describe this with the term "loss of abstraction".

3 Limitations in Decision Support: State of the Art

Scheduling applications should incorporate knowledge provided by the human expert.

In addition to the *input* of preferences and modifications, they should allow their *development* and *refinement*, too [20, 23]. Furthermore, the decisions about *how* identified preferences and modifications are incorporated should be left to the human.

Human scheduling activity is considered as a problem-solving or decision-making activity [22]. We examine the human decision making process and how it is supported with the present state of the art.

3.1 Making Decisions

Decision-making in General. Scheduling can be modeled as decision process consisting of *intelligence*, *design* and *choice* [7, 9]. The intelligence phase involves the recognition of the problem at the start of the decision process. After that, possible solutions are evaluated in the design phase. The best alternative is finally selected in the choice step. We add a *completion* step, if the selected solution yet has to be completed. If the completion step is still complex, a new decision process is triggered. The decision processes are chained that way until the task is accomplished.

The decision process is influenced by skills and knowledge of the human. We distinguish skill-based (SBB), rule-based (RBB) and knowledge-based reasoning (KBB) [4]. KBB involves the whole decision process. RBB is applied, if familiar patterns in the data map to a rule that implies a certain action. In SBB, Perception is mapped to action directly. As shown in Fig. 2 RBB and SBB shorten the decision process.

Decision-making in Scheduling. The decision stages can be applied directly to human scheduling activities.

Intelligence: The human operator recognizes need for action. It is important for him to collect and interpret the data of schedules in order to find required preferences or

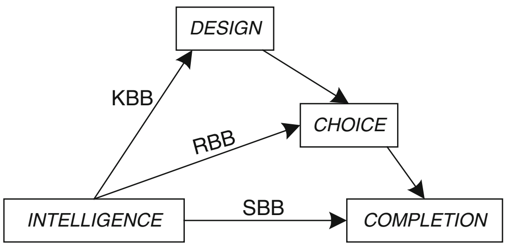


Fig. 2. Stages in decision-making and shortcuts.

modifications. Having identified them, he either intervenes in the process of schedule creation or modifies an existing schedule.

Design: In the design stage the human operator compares alternative solutions that fulfill his preferences. Depending on the abstraction level this involves comparing

- different schedules (level 1)
- different assignments of jobs to resources (level 2)
- different orders of jobs within a resource (level 2)
- different scheduling properties of jobs (level 3)

Each considered alternative is evaluated with regard to optimality and practicability. However, only valid schedules can be evaluated. Due to the earlier mentioned "loss of abstraction" the human operator has to make decisions about the details below the abstraction level of the task. This leads to a new decision process in order to find a valid implementation of the solution to be considered [11]. The original decision process is compromised, as the human must keep track of nested design stages at different levels.

Choice and Completion: The human operator chooses the best suited schedule. If complete schedules are compared in the design stage the completion step can be omitted.

Types of Reasoning in Scheduling. **KBB** contains the three stages intelligence, design and choice. It depends both on the experience of the human operator and on the characteristics of the task whether the decision process can be shortened by SBB or RBB.

SBB and RBB: Applies, if the human operator deals with the task repeatedly or if there are best practices, such that the best suited alternative is known from experience. In this case the dispatcher does not evaluate alternative solutions.

3.2 Human–Computer-Interaction for Decision Support

Scheduling systems usually offer two contrary interaction models: manual or fully automatic scheduling [2, 14]. In the following sections possible configurations of these models are analyzed with regard to the decision support they provide to the human dispatcher.

Manual Scheduling. Manual optimization of schedules is a monotonous job unsuitable to humans [8]. Due to the structure of the problems the number of valid positions for jobs is exponential [1] which makes it difficult for the human to find the optimal costs. In manual scheduling, the human is at level 3 in the abstraction hierarchy (Fig. 1). He has to fulfill all constraints, which is too difficult to simultaneously optimize with regard to objective functions. For this reason, it is often too complex for the user to produce several alternative solutions in the course of KBB [15].

Automated Scheduling with Subsequent Modifications. In order to avoid a recreation of the whole schedule, it is common practice to use an existing schedule as starting point. It is either automatically created from a set of unscheduled jobs or reused from the past. The manual modifications applied to the schedule

1. suffer from the "loss of abstraction"
2. destroy the global optimality of the initial schedule (with regard to a certain objective function), if it was produced automatically.

Automated Scheduling with User-defined Constraints and Parameters. Another common approach in dealing with schedule modifications is a repeated re-optimization of the schedule with additional user-defined constraints that were added to the formal scheduling model. The underlying solver takes the human's modifications into account and produces a schedule that is optimal with regard to the selected objective function.

This approach is useful for SBB or RBB, if no judgment of the human is needed to decide about the best way to implement the decision. Despite of this, it does not always provide an optimal decision support for the dispatcher:

1. The changes in the schedule that are caused by the re-optimization often affect more jobs, than those that belong to the dispatcher's decision. This makes it difficult to compare the schedules before and after the optimization step. However, only by a comparison, the dispatcher is able to assess the impact of his decision on the schedule.
2. The optimization does not offer enough alternatives to the human. Usually the dispatcher can choose among several objective functions or influence weights and parameters of optimization algorithms [2]. Thereby he is at level 1 in the abstraction hierarchy (Fig. 1). At this level the comparison of alternatives for lower-level decisions about assignments, orders and start-times of certain jobs in the course of KBB is difficult [16]. The dispatcher can produce alternatives by changing user-defined constraints, but because of 1. the results are often incomprehensible.

4 Hypothesis for Optimal Decision Support

4.1 Temporary Constraints for the Support of Preferences

Modifications to the scheduling problem can be applied, for example, by adding or removing jobs, resources or constraints among them. They change the solution space, such that, for example, some of the former variable assignments may become infeasible. In contrast, preferences do not change the solution space, but they filter the current solution space by certain criteria. However, finding the preferred solution is often not possible, because

– The solution space is often so large, that the user cannot browse it. Humans have cognitive limits that restrict their ability to compare alternative solutions [16].

– Scheduling applications usually do not provide an unlimited set of alternative solutions. Therefore the desired solution has to be produced manually (Sect. 3.2).

For this reason we suggest that there should be possibilities to define *temporary* constraints that express a preference the user wants to implement. They are not to override existing constraints permanently, but to temporarily enforce certain values from the domain of a scheduling variable. The system should treat them as normal constraints until the user removes them. This way every automatic solution has the desired properties, giving the user the chance to compare different scenarios by successively changing the temporary constraints.

4.2 The User's Abstraction Levels

It is an important issue for decision support to keep the human operator at the level of abstraction, that is related to his preference and to allow him the free choice of the level of automation [13].

Level 3. This level requires the least amount of automation. In contrast to manual scheduling as described above, the system should supervise the compliance with the underlying constraints to prevent faulty decisions. In doing so it is not sufficient to show an error message as soon as a constraint is violated. We rather suggest to visualize the scope of action already when the human is about to make a decision. According to the types of constraints in Sect. 2.1 this means highlighting valid properties for the considered job that

1. meet its individual constraints
2. meet its constraints in relation to other jobs

with regard to the state of the current schedule. This way the human does not have to make the effort to withdraw a faulty decision.

In addition to highlighting the current scope of action, the system should provide feedback in case the schedule has become infeasible [17]. We propose it should indicate, at which level of abstraction the schedule can be made feasible. For example, if the user changes the start time of a job, such that it overlaps with his successor, the system should inform the user whether the conflict can be resolved within the same resource or whether one or more jobs must be reassigned to different resources in order to use this start time. This kind of system feedback makes the consequences of a scheduling decision clear. It prevents the user from unsuccessful attempts to locally adapt a schedule, if instead global changes are necessary (or if the decision cannot be implemented at all without constraint relaxation). It can be implemented by an internal pre-calculation of possible states of the schedule.

Furthermore, the consequences with regard to the quality should be visualized for instance by updating the results of optimization functions.

Level 2. Properties that are not fixed to a certain value are to be automatically resolved by the computer. The human makes decisions on *some selected* scheduling properties of either individual jobs or groups of jobs [22]. The computer is required to

Table 1. Properties of selected jobs assigned by human and computer at different sublevels of level 2.

Sublevel	Human	Computer
Sequencing	Resource, relative position, cost function	Start time
Assignment	Resource, cost function	Relative position, start time
Quality	Cost function	Resource, relative position, start time

solve the remaining properties such that all constraints are satisfied and the schedule is optimal or at least good with regard to the cost function.

In doing so the properties should be processed corresponding to the abstraction hierarchy. The computer resolves properties *below* the user's abstraction level. Thus the portion of work of the computer increases with the sublevels as shown in Table 1.

The degree, to which the computer can reschedule selected jobs, consequently depends from the chosen abstraction level:

– Sequencing level: the initial order of the jobs in this resource must be preserved as much as possible
– Assignment level: the order of the jobs can only be changed within the selected resource
– Quality level: selected jobs can be assigned to different resources

If partial optimization takes place for *selected* jobs or groups of jobs, changes to the *remaining* jobs in the schedule are often necessary, too. We recommend that remaining jobs should only be changed until *feasibility* of the overall schedule is achieved. Changes, that are necessary in order to achieve global *optimality*, should not be made. This way the user can focus on the scheduling of his selected jobs without being distracted by side-effects on further jobs.

Level 1. Full automation is applied at this level. The human operator is only concerned about the cost function the computer should use to optimize the whole schedule. Preferences and modifications are taken into account. We refer to [18] for an approach to visualize global quality indicators.

4.3 Decision-Making at Intermediate Abstraction Levels

Both modifications (constraints) and preferences (temporary constraints) can refer to *any* abstraction level. With the help of semi-automatic interaction models at level 3 and 2, the user can decide how much he participates in the implementation of a preference or modification and how much the current schedule is allowed to be changed. For example, if the user has a preference about the start time of a job, after defining the corresponding constraint he has several options:

– Manual scheduling at level 3: the system visualizes the resources that are able to take the job at the desired time; the user adapts the schedule manually.

– Semi-automatic scheduling at level 2: the user decides for a certain resource and if necessary makes precedence decisions; the system completes the scheduling process.
– Fully automatic scheduling at level 3.

At each level, the user is free to try different scheduling scenarios, for example, different resources at level 2 or different goal functions at level 1. This way KBB is supported. KBB and RBB are also supported: The user is able to define precisely how the current schedule should be adapted. This is in contrast to full automation as described in Sect. 3.2, where side-effects on the overall schedule could occur.

5 Design of Interactive Scheduling Interfaces

5.1 Interactive Decision Support Features

We have designed a set of interaction features that can be used to build a scheduling interface providing the recommended decision support. They are shown in Fig. 3. We focus on scheduling-specific features and neglect commonly used features like Undo/Redo, as they can be found in the standard literature about successful user interface design [6].

Fixation (FIX). The interface allows the direct input of preferences for one or more jobs regarding the start time, order or resource allocation. They are turned into additional temporary constraints to be considered by all features. Fixations should be easily applicable and removable. For example, in order to fix the start time and resource of a job, we suggest interactive pins, that can be added to and removed from a job.

Level 3: Constraint Highlighting (CH): We use colors to visualize the domain of the property of a job in the current schedule [24]. The interface recognizes the intention to change a property of a job and colors possible values

– *red,* if they are invalid,
– *green,* if they are valid with regard to constraints of *the individual job*.

Level 3: Enhanced Constraint Highlighting (ECH): Additional to CH: values of properties that violate constraints *in relation to other jobs of the same resource* are colored

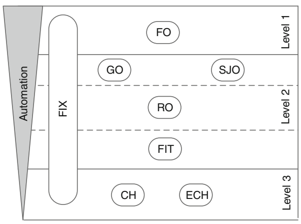


Fig. 3. Decision support features at different abstraction levels.

– *yellow,* if the value can be applied as soon as the properties of conflicting jobs are adapted
– *grey,* if the value can never be applied in conjunction with the conflicting jobs.

This way the colors indicate the minimum level, at which a preference or modification can be implemented:

– *green:* level 3 or higher
– *yellow:* sequencing level 2 or higher
– *grey:* quality level 2 or higher.

In order to avoid the loss of abstraction, the user can have the schedule optimized automatically at the desired level. For this we design the following optimization features:

Level 2 Sequencing: Fit-in (FIT): The interface allows the user to define the position of a job within the sequence. As soon as the user starts the FIT-action, the current position of the job is automatically turned into a temporary precedence constraint. The system then looks for a valid start time for the job.

Level 2 Assignment: Resource Optimization (RO): The user can select single resources and optimize their schedules independently from one another. He can choose from different goal functions, which are available in the application.
Level 2: Quality: Single Job Optimization (SJO): The interface allows selecting a single job in the schedule and triggers automatic optimization of its position. The user can choose from different goal functions.

Level 2 Quality: Group Optimization (GO): Like SJO. Any group of jobs from different resources can be selected.

Level 1: Full Optimization (FO): A control to optimize the whole schedule. It allows choosing from various built-in cost functions.

5.2 Example Interfaces

Our hypothesis can be applied to different graphical interfaces. The information visualization should follow the principles of Ecological Interface Design [6, 10] and display information according to the abstraction hierarchy. We show two example interfaces that include our recommended decision support features.

Fleet Scheduling (Gantt Chart Based). The interface used in our experiments is sketched in Fig. 4. We decided to use a Gantt Chart, as it clearly shows the sequence of jobs in time and the travel times between them. This makes it easy to analyze start times and resources of jobs in order to derive certain preferences. For further support we provide a map.

The human operator can move the jobs per Drag and Drop. If he starts dragging (enhanced) constraint highlighting is applied to the background of the Gantt chart: A

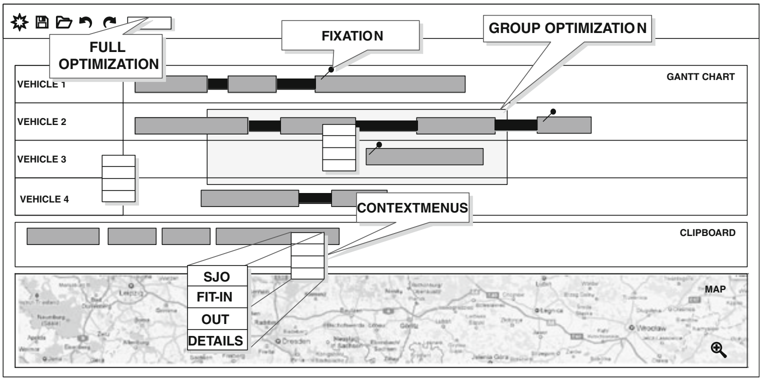


Fig. 4. Interface design for vehicle routing.

job can be dropped at any time slot colored green or yellow, in the latter case the FIT feature can be used to put the job correctly in the sequence.

Furthermore, SJO, RO and GO are available through context menus and provide the two cost functions "minimize traveltime" and "balance workload". For example, if a group of jobs is optimized with the criteria "balance workload", the jobs are reassigned to the resources, such that the workload is balanced between all resources.

Scheduling modifications and preferences can be defined in property dialogs and by using the pin (FIX) that temporarily fixes both start time and resource of a job. A button to create schedules from scratch (FO) is also provided.

Nurse Rostering (Shift Schedule Based). A possible interface for nurse rostering is shown in Fig. 5. Each shift requires a certain number of nurses (according to the terminology used in this paper shifts are "jobs" and nurses "resources") that must be included. The cost function usually deals with considering the preferences of the individual nurses.

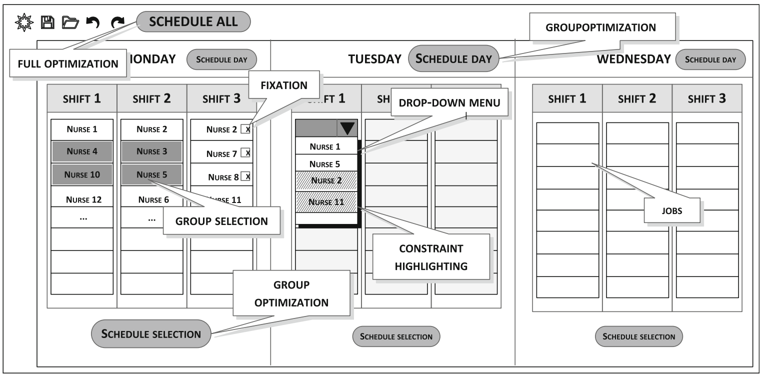


Fig. 5. Interface design for nurse rostering.

Nurses can be assigned to shifts with the help of a drop-down menu, whose entries are colored according to CH and ECH. For example, if a nurse had a night shift the day before it must not be assigned to the early shift due to legal requirements. In this case the nurse is colored grey in the drop down menu. Her assignment requires rescheduling of the day before. Furthermore the interface contains features to select a group of nurses (SJO, GO) or the whole schedule (FO) for automatic optimization. In this case fixed nurses (FIX) are kept unchanged.

6 Evaluation of the Decision Support

6.1 Setup of the Usability Test

Combining Decision Support Features to Interaction Models. In order to prove our claims from Sect. 1 it remains to provide an empirical evaluation of the suitability of the features for performing scheduling tasks at different abstraction levels. We also want to measure the quality that can be achieved in terms of the cost function.

For this we combine the features to 5 interaction models located at different abstraction levels. Models 0–4 correspond to the models described in Sect. 3.2. Model 5 contains the decision support features from Sect. 5 (except from FO):

Model 0: manual scheduling without decision support
Model 1/2: manual scheduling at level 3 (Model1: CH, Model2: ECH)
Model 3: FO at level 1, subsequent manual modifications at level 3 are allowed, fixation is not allowed (FO + ECH)
Model 4: like model 3, fixation is allowed (FIX + FO + ECH)
Model 5: supports all abstraction levels (except from level 1) (FIX + ECH + FIT + SJO + RO + GO).

Several test tasks with scheduling preferences at different abstraction levels are carried out by peer groups. Each model is used for each task.

Test Groups. We have formed test groups each consisting of 7 students from different faculties of our institution. The subjects were asked to perform 6 scheduling tasks as peer groups for each Model 0–5. The models available for the particular tasks were dependent on the test group. We evaluate by the average performance and confidence interval in the following metrics: accumulated travel time, task completion, time effort, number of undo operations and number of manual interactions. The tests took 3 h per participant including a briefing of 30 min at the start. The maximum duration for each task was set to 15 min.

Design of the Test Tasks. The participants had no experiences in scheduling. Therefore the relevant scheduling preferences that would otherwise arise from the expert knowledge of the scheduler had to be predefined for each task. The participants had to solve problems like:

1. Schedule a set of jobs such that the total travel time is minimized. For some jobs there are precedence constraints **(level 2 sequencing)**.

2. An additional vehicle is to be utilized. Change the given schedule such that some suitable jobs are assigned to it **(level 2 assignment)**.
3. An event occurs and requires an additional job. The working schedule must include the job as early as possible, but has to remain unchanged until 10 o'clock **(level 3)**.

The tasks are to be carried out with 4 vehicles and about 25 predefined jobs. All jobs have time window and resource constraints.

6.2 Results

Usability Metric 1: Accumulated Travel Time. The overall ranking of the models is shown in Fig. 6 (1 is the best, 6 the worst rank). It confirms the assumption that models 4 and 5 generally provide the best decision support. In average, model 4 produced a slightly better quality than model 5. This can be explained by the fact, that model 5 encourages local optimization steps (SJO, RO, FIT…), while model 5 performs a global optimization. Model 4 is intended to provide better decision support; consequently the global optimality has to be sacrificed in favor of a greater comprehensibility of the results.

The schedules created with level 3-features only were worse than those created with higher-level-features. This confirms the assumption that skill-based scheduling tasks not requiring human judgement should be carried out by the computer. CH and ECH help the human to find a scheduling decision for *some* jobs, but are not sufficient for creating *complete* schedules.

Comparing models 3 and 4, the quality decreases if fixation is not allowed. This suggests that preferences should be incorporated in advance (FIX) rather than after automated optimization, in order to achieve the optimal quality.

Usability Metric 2: Task Success. The number of participants that have managed to obtain a solution is shown in Fig. 7. A task was considered successful, if the schedule did not violate any time window or resource constraints and the scheduling preferences were fulfilled.

With models 1, 2 and "None" many participants ran into dead-ends, where they were not able to insert further jobs in the clipboard. In this case model 2 merely

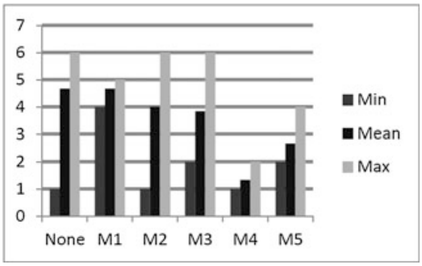


Fig. 6. Ranking of the models averaged over the tasks.

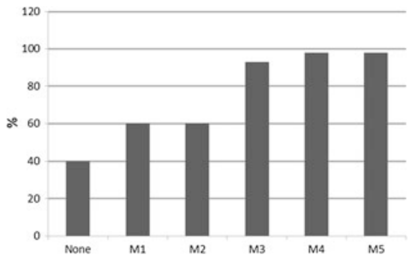


Fig. 7. Rate of successful task completion.

depicted a grey Gantt chart background. They would have to manually backtrack former decisions. However, testers would rather give up at this point.

Usability Metric 3: Task Duration. The average time, users required to solve the tasks (deadline was 15 min) is shown in Fig. 8. Although the time needed with no model is particularly high, in general the models have a high variance in their execution time. How much time a test person spent to fulfill a task was strongly dependent on his motivation and ideas to improve the schedule. The runtime of the system to solve the scheduling problem was negligible.

Usability Metric 4: Interaction Frequency. Figure 9 shows the number of undo operations averaged over the number of participants. Models 4 and 5 have a strikingly high occurrence of undo, which refers to the general behavior in the design phase, if

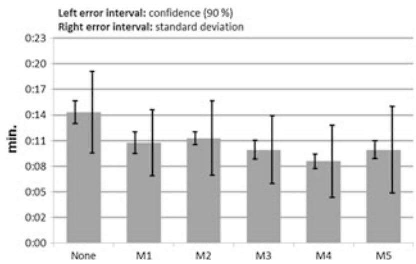


Fig. 8. Average task duration.

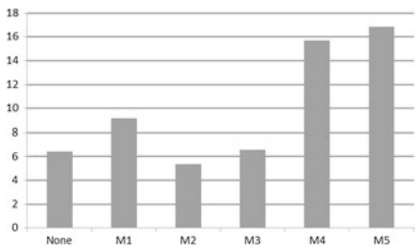


Fig. 9. Average number of undo operations.

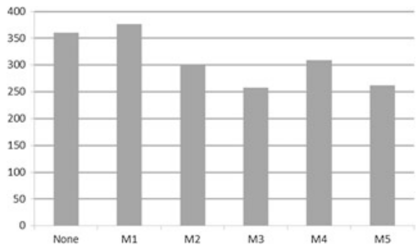


Fig. 10. Average number of manual operations.

there are high-level scheduling features. It consists of alternately applying and reversing automated scheduling features until a satisficing solution is found. Model 1 has a small peak in undo-operations, as there is no aid to predict if an operation will be feasible. Model 2 compensates for this with the background-color grey.

Figure 10 shows the average number of manual operations (drag and drop of jobs). As expected the manual effort is the higher, the less support is provided. However, manual scheduling is not completely replaced by automated features, as the user-performs subsequent changes or sets certain jobs according to his ideas.

7 Conclusions

We proposed 8 interaction features to enhance human interaction in scheduling. These features were evaluated in a quantitative study (usability test) with regard to 4 relevant metrics. The results are:

1. The practicability of resulting schedules improves with features to manually fixate, reorder and optimize groups of jobs.
2. The success rate (solved tasks in given time) is highly influenced by the availability of automated scheduling features.
3. Automated scheduling features encourage the user to explore his scope of action on the basis of trial and error (optimize - undo).

References

1. Burke, E.K., Kendall, G.: Search methodologies: introductory tutorials in optimization and decision support techniques. Springer, Heidelberg (2005)
2. Fransoo, J.C., Waefler, T., Wilson, J.: Behavioral Operations in Planning and Scheduling. Springer, Berlin (2011)
3. Kallehauge, B., Larsen, J., Madsen, O., Solomon, M.: Vehicle routing problem with time windows. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.) Column Generation, pp. 67–98. Springer, New York (2005)
4. Rasmussen, J.: Skills, rules, and knowledge; signals, signs and symbols, and other distinctions in human performance models. IEEE Trans. Syst. Man Cybern. **13**(3), 157–266 (1983)

5. St-Cyr, O., Burns, C.M.: Mental models and the abstraction hierarchy: assessing ecological compatibility. In: Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting, pp. 297–301 (2001)
6. Shneidermann, B.: Designing the User Interface. Pearson Addison-Wesley, Upper Saddle River (2010)
7. Burstein, F., Holsapple, C.W. (eds.): Handbook on Decision Support Systems 1 and 2. Springer, Berlin (2008)
8. Burstein, M.H., McDermott, D.V.: Issues in the development of human-computer mixed initiative systems. Cognit. Technol. 285–303 (1997)
9. Turban, E., Sharda, R., Delen, D.: Decision Support and Business Intelligence Systems. Prentice Hall, Pearson (2010)
10. Van Westrenen, F.: Cognitive work analysis and the design of user interfaces. Cognit. Technol. Work **13**(1), 31–42 (2011)
11. Wezel, W., Jorna, R., Meystel, A. (eds.): Planning in Intelligent Systems. Wiley Interscience, New York (2006)
12. Snoo, C.D., van Wezel, W., Jorna, R.J.: An empirical investigation of scheduling performance criteria. J. Operat. Manag. **29**(3), 181–193 (2011)
13. Johnson, K., Ren, L., Kuchar, J., Oman, C.: Interaction of automation and time pressure in a route replanning task. In: Proceedings HCI-02. AAAI (2002)
14. Parasuraman, R., Sheridan, T.B., Wickens, C.D.: A model for types and levels of human interaction with automation. Trans. Syst. Man Cybern. Part A **30**(3), 286–297 (2000)
15. Cegarra, J.: A cognitive typology of scheduling situations: a contribution to laboratory and field studies. Theor. Issues Ergon. Sci. **9**(3), 201–222 (2008)
16. Cegarra, J., Gacias, B., Lopez, P.: Implications of technological changes in vehicle routing interfaces for planners' constraint processing. Hum. Factors Ergon Manuf. Serv. Ind. **22**(5), 468–480 (2012)
17. Freuder, E.C., Wallace, R.J., Nordlander, T.E.: Debugging constraint models with metamodels and metaknowledge. Cork Constraint Computation Center and Department of Computer Science, Cork, Ireland (2002)
18. Lambeck, C., Schmalzried, D., Alt, R., Groh, R.: Changing concepts in human-computer interaction in real-time enterprise systems. In: Proceedings of the 14th International Conference on Enterprose Information Systems, ICEIS (2012)
19. Topi, H., Lucas, W., Babaian, T.: Identifying usability issues with an ERP implementation. In: Proceedings of the International Conference on Enterprise Information Systems (ICEIS), pp. 128–133 (2005)
20. Bainbridge, L.: The change in concepts needed to account for human behavior in complex dynamic tasks. IEEE Trans. Syst. Man. Cybern. Part A: Syst. Hum. **27**(3), 351–359 (1997)
21. Jackson, S., Wilson, J.R., MacCarthy, B.L.: A new model of scheduling in manufacturing: tasks, roles, and monitoring. Hum. Factors **46**(3), 533–550 (2004)
22. Hoc, J., Guerin C., Mebarki, N.: The nature of expertise in scheduling: the case of timetabling. Hum. Factors Ergon. Manuf. Serv. Ind. (in press) (2012)
23. Kazakci, A.O., Tsoukias, A.: Designing or Planning? - Elements for a cognitive foundation of design aiding. Lamsade – CNRS, Université Paris Dauphine (2002)
24. Dillan, S., Ringwelski, G.: Using constraint propagation to improve the usability of an intelligent decision support system. In: Proceedings of the International Workshop on Innovative Scheduling and other Applications using CP-AI-OR (ISA 2011). Computer Science Report 01/11 of Brandenburg University of Technology Cottbus, (2011)

Combining Learning Algorithms: An Approach to Markov Decision Processes

Richardson Ribeiro[1(✉)], Fábio Favarim[1], Marco A. C. Barbosa[1], Alessandro L. Koerich[2], and Fabrício Enembreck[2]

[1] Graduate Program in Computer Engineering, Federal Technological University of Paraná, Pato Branco, Paraná, Brazil
{richardsonr, favarim, marcocb}@utfpr.edu.br
[2] Post-Graduate Program in Computer Science, Pontificial Catholical University of Paraná, Curitiba, Paraná, Brazil
{koerich, enembreck}@ppgia.pucpr.br

Abstract. In this paper we present a technique for estimating policies which combines instance-based learning and reinforcement learning algorithms in Markovian environments. This approach has been developed for speeding up the convergence of adaptive intelligent agents that using reinforcement learning algorithms. Speeding up the learning of an intelligent agent is a complex task since the choice of inadequate updating techniques may cause delays in the learning process or even induce an unexpected acceleration that causes the agent to converge to a non-satisfactory policy. Experimental results in real-world scenarios have shown that the proposed technique is able to speed up the convergence of the agents while achieving optimal policies, overcoming problems of classical reinforcement learning approaches.

Keywords: Reinforcement learning · Dynamic environments · Adaptive agents

1 Introduction

Markov Decision Processes (MDP) are a popular framework for sequential decision-making for single agents, when agents' actions have stochastic effect on the environment state and need to learn how to execute sequential actions. Adaptive intelligent agents emerge as an alternative to cope with several complex problems including control, optimization, planning, manufacturing and so on. A particular case is an environment where events and changes in policy may occur continuously (i.e., dynamic environment). A way of addressing such a problem is to use Reinforcement Learning (RL) algorithms, which are often used to explore a very large space of policies in an unknown environment by trial and error. It has been shown that RL algorithms, such as the Q-Learning algorithm [1], converge to optimal policies when a large number of trials are carried out in stationary environments [2].

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 172–188, 2013.
DOI: 10.1007/978-3-642-40654-6_11,

Several works using RL algorithms and adaptive agents in different applications can be found in the literature [3–9]. However, one of the main drawbacks of RL algorithms is the rate of convergence which can be too slow for many real-world problems, e.g. traffic environments, sensor networks, supply chain management and so forth. In such problems, there is no guarantee that RL algorithms will converge, since they were originally developed and applied to static problems, where the objective function is unchanged over time. However, there are few real-world problems that are static, i.e. problems in which changes in priorities for resources do not occur, goals do not change, or where there are tasks that are no longer needed. Where changes are needed through time, the environment is dynamic.

In such environments, several approaches for achieving rapid convergence to an optimal policy have been proposed in recent years [10–13]. They are based mainly on the exploration of the state-action space, leading to a long learning process and requiring great computational effort.

To improve convergence rate, we have developed an instance-based reinforcement learning algorithm coupled with conventional exploration strategies such as the ε-greedy [14]. The algorithm is better able to estimate rewards, and to generate new action policies, than conventional RL algorithms. An action policy is a function mapping states to actions by estimating a probability that a state s' can be reached after taking action a in state s.

In MDP, algorithms attempt to compute a policy such that the expected long-term reward is maximized by interacting with an environment [2]. The approach updates into state-action space the rewards of unsatisfactory policies generated by the RL algorithm. States with similar features are given similar rewards; rewards are anticipated and the number of iterations in the Q-learning algorithm is decreased.

In this paper we show that, even in partially-known and dynamic environments, it is possible to achieve a policy close to the optimal very quickly. To measure the quality of our approach we use a stationary policy computed previously, comparing the return from our algorithm with that from the stationary policy, as in [15].

This article is organized as follows: Section 2 introduces the RL principles and the usage of heuristics to discover action policies. The technique proposed for dynamic environments is presented in Sect. 3 where we also discuss the Q-Learning algorithm and the k-Nearest Neighbor (k-NN) algorithm. Section 4 gives experimental results obtained using the proposed technique. In the final section, some conclusions are stated and some perspectives for future work are discussed.

2 Background and Notation

Many real-life problems such as games [16,17], robotics [18], traffic light control [19,20] or air traffic [21,22], occur in dynamic environments. Agents that interact in this kind of environment need techniques to help them, e.g., to reach some goal, to solve a problem or to improve performance. However because individual circumstances are so diverse, it is difficult to propose a generic approach (heuristics) that can be used to deal with every kind of problem. Environment is the world in which an agent operates.

A dynamic environment consists of changing surroundings in which the agent navigates. It changes over time independent of agent actions. Thus, unlike the static case, the agent must adapt to new situations and overcome possibly unpredictable obstacles [23,24]. Traditional planning systems have presented problems when dealing with dynamic environments. In particular, issues such as truth maintenance in the agent's symbolic world model, and replanning in response to changes in the environment, must be addressed.

Predicting the behavior (i.e., actions) of an adaptive agent in dynamic environments is a complex task. The actions chosen by the agent are often unexpected, which makes it difficult to choose a good technique (or heuristic) to improve agent performance. A heuristic can be defined as a method that improves the efficiency in searching a problem solution, adding knowledge about the problem to an algorithm.

Before discussing related work, we introduce the MDP which is used to describe our domain. A MDP is a tuple $(S, A, \partial^a_{s,s'}, R^a_{s,s'}, \gamma)$ where S is a discrete set of environment states that can be composed by a sequence of state variables $< x_1, x_2, ..., x_y >$. An episode is a sequence of actions $a \in A$ that leads the agent from a state s to s'. $\partial^a_{s,s'}$ is a function defining the probability that the agent arrives in state s' when an action a is applied in state s. Similarly, $R^a_{s,s'}$ is the reward received whenever the transition $\partial^a_{s,s'}$ occurs and $\gamma \in \{0...1\}$ is a discount rate parameter.

A RL agent must learn a policy $Q : S \rightarrow A$ that maximizes its expected cumulative reward [1], where $Q(s, a)$ is the probability of selecting action a from state s. Such a policy, denoted as Q^*, must satisfy Bellman's equation [14] for each state $s \in S$ (Eq. 1).

$$Q(s, a) \leftarrow R(s, a) + \gamma \sum_{s'} \partial(s, a, s') \times \max Q(s', a) \tag{1}$$

where γ weights the value of future rewards and $Q(s, a)$ is the expected cumulative reward given for executing an action a in state s. To reach an optimal policy (Q^*), a RL algorithm must iteratively explore the space $S \times A$ updating the cumulative rewards and storing such values in a table $\hat{Q}$.

In the Q-learning algorithm proposed by Watkins [1], the task of an agent is to learn a mapping from environment states to actions so as to maximize a numerical reward signal. The algorithm approaches convergence to Q^* by applying an update rule (Eqs. (2), (3)) after a time step t:

$$v \leftarrow \gamma \max Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t) \tag{2}$$

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha[R(s_t, a_t) + v] \tag{3}$$

where V is the utility value to perform an action a in state s and $\alpha \in \{0,1\}$ is the learning rate.

In dynamic environments such as traffic jams, it is helpful to use strategies like ε-greedy exploration [14] where the agent selects an action with the greatest Q value with probability $1-\varepsilon$. In some Q-Learning experiments, we have found that the agent does not always converge during training (see Sect. 4). To overcome this problem we have used a well known Q-Learning property: actions can be chosen using an exploration strategy. A very common strategy is random exploration, where an action is randomly chosen with probability ε and the state transition is given by Eq. 4.

$$Q(s) = \begin{cases} \max Q(s,a), & \text{if} \quad q > \varepsilon \\ a_{random}, & \text{otherwise} \end{cases} \tag{4}$$

where q is a random value with uniform probability in $[0,1]$ and $\varepsilon \in [0,1]$ is a parameter that defines the exploration trade-off. The greater the value of ε, the smaller is the probability of a random choice, and a_{random} is a random action selected among the possible actions in state s.

Several authors have shown that matching some techniques with heuristics can improve the performance of agents, and that traditional techniques, such as ε-greedy, yield interesting results [10,11,25]. Bianchi [11] proposed a new class of algorithms aimed at speeding up the learning of good action policies. An RL algorithm uses a heuristic function to force the agent to choose actions during the learning process. The technique is used only for choosing the action to be taken, while not affecting the operation of the algorithm or modifying its properties. Bianchi et al. [11] also proposed an automatic method based on heuristic propagation to compute a policy in runtime from the structure extracted from the domain. The technique propagates from its goal-state, the correct actions which would lead to that state. Drummond [25] proposes a method that accelerates RL by transferring parts of previously learned solutions to a new problem, exploiting the results of prior learning to speed up the process. The method is called *reusing policies*, where the heuristics based on the cases are used as the state transition rules to choose the a_t action, taken in the s_t state. Then, ε-greedy random exploration strategies are used to estimate the T transition probability functions and the R reward.

Butz [26] proposes the combination of an online model learner with a state value learner in a MDP. The model learner learns a predictive model that approximates the state transition function of the MDP in a compact, generalized form. State values are evaluated by means of the evolving predictive model representation. In combination, the actual choice of action depends on anticipating state values given by the predictive model. It is shown that this combination can be applied to increase further the learning of an optimal policy.

Koenig and Simmons [27] proposed a method that stores the reinforcements received during the learning of an agent in a separated structure. This structure has a table storing the reinforcements received in each state and a threshold is used to generate a map of the positions where the agent received more reinforcements. Therefore, the tables that keep the positive and negative reinforcements contain information for defining the heuristics, together with information about the actions to be carried out.

Bianchi et al. [28] improved action selection for online policy learning in robotic scenarios combining RL algorithms with heuristic functions. The heuristics can be used to select appropriate actions, so as to guide exploration of the state-action space during the learning process, which can be directed towards useful regions of the state-action space, improving the learner behavior, even at initial stages of the learning process.

In this paper we propose going further in the use of exploration strategies to achieve a policy closer to the Q^*. To do this we have used policy estimation techniques based on an instance learning, such as the k-Nearest Neighbors (k-NN) algorithm. We have observed that is possible to reuse previous states, eliminating the need of a prior heuristic.

3 *k*-NR: Instance-Based Reinforcement Learning Approach

In RL, learning takes place through a direct interaction of the algorithm with the agent and the environment. Unfortunately, the convergence of the RL algorithms can only be reached after an exhaustive exploration of the state-action space, which usually converges very slowly. However, the convergence of the RL algorithm may be accelerated through the use of strategies dedicated to guiding the search in the state-action space.

The proposed approach, named k-Nearest Reinforcement (k-NR), has been developed from the observation that algorithms based on different learning paradigms may be complementary to discover action policies [29]. The information gathered during the learning process of an agent with the Q-Learning algorithm is the input for the k-NR. The reward values are calculated with an instance-based learning algorithm. This algorithm is able to accumulate the learned values until a suitable action policy is reached.

To analyze the convergence of the agent with the k-NR algorithm, we assume a generative model governing the optimal policy. With such a model it is possible to evaluate the learning table generated by the Q-Learning algorithm. To do this, an agent is inserted into a partially known environment with the following features:

1. Q-Learning algorithm: learning rate (α), discount factor (γ) and reward (r);
2. Environment E: the environment consists of a state space where there is an initial state ($s_{initial}$), a goal state (s_{goal}) and a set of actions A=$\{\uparrow,\downarrow,\rightarrow,\leftarrow\}$, where $\uparrow$, $\downarrow$, $\rightarrow$, $\leftarrow$ mean respectively *east*, *south*, *north* and *west* (Fig. 1).

A state s is an ordered pair (x, y) with positional coordinates on the axis X and Y respectively. In other words, the set of states S represents a discrete city map. A status function $st : S \rightarrow ST$ maps states and traffic situations where $ST = \{(-0.1, -0.2, -0.3, -0.4, -1.0, 1.0)\}$, where $-0.1, -0.2, -0.3, -0.4, -1.0$

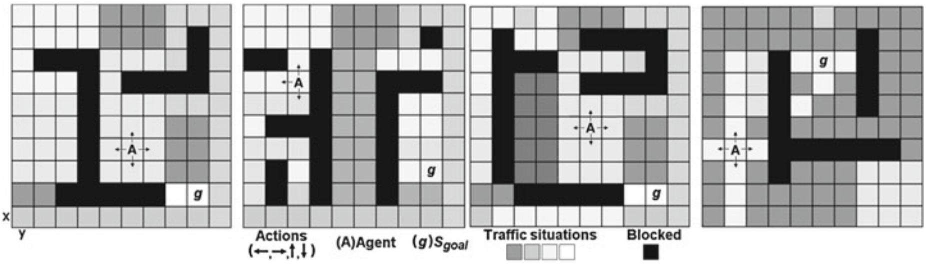


Fig. 1. Environment: A single agent is placed at random positions in the grid, having a visual field depth of 1.

and 1.0 mean respectively *free*, *low jam*, *jam* or *unknown*, *high jam*, *blocked*, and *goal*. After each move (transition) from state s to s' the agent knows whether its action is positive or negative through the rewards attributed by the environment. Thus, the reward for a transition $\partial^a_{s,s'}$ is $st(s')$ and Eqs. (2), (3) is used as update function. In other words, the agent will know if its action has been positive if, having found itself in a state with traffic jam, its action has led to a state where the traffic jam is less severe. However, if the action leads the agent to a more congested status then it receives a negative reward.

The pseudocode to estimate the values for the learning parameters for the Q-Learning using the k-NR is presented in Algorithm 1. The following definitions parameters are used in such an algorithm:

- a set $S = \{s_1, \ldots, s_m\}$ of states;
- an instant discrete steps $step = 1, 2, 3, ..., n$;
- a time window T_x that represents the learning time (cycle(x) of steps);
- a set $A = \{a_1, \ldots, a_m\}$ of actions, where each action is executable in a step n;
- a status function $st : S \to ST$ where $ST = \{-1, -0.4, -0.3, -0.2, -0.1\}$;
- learning parameters: α=0.2 and γ=0.9;
- a learning table $QT : (S \times A) \to \mathbb{R}$ used to store the accumulative rewards calculated with the Q-Learning algorithm;
- a learning table $kT : (S \times A) \to \mathbb{R}$ used to store the reward values estimated with the k-NN;
- *#changes* is the number of changes in the environment.

3.1 *k*-NN and *k*-NR

The instance-based learning paradigm determines the hypothesis directly from training instances. Thus, the k-NN algorithm saves training instances in the memory as points in an n-dimensional space, defined by the n attributes which describe them [30,31]. When a new instance must be classified, the most frequent class among the k nearest neighbors is chosen. In this paper the k-NN algorithm is used to generate intermediate policies which speed up the convergence of RL algorithms. Such an algorithm receives as input a set of instances

Algorithm 1: Policy estimation with k-NR.

Require: Learning Table: $QT(s, a)$;
$kT(s, a)$; S=$\{s_1, ..., s_m\}$; A=$\{a_1, ..., a_m\}$
st: S $\rightarrow$ ST;
Time window T_x;
Environment E;
Ensure:
1. **for all** $s \epsilon$ S **do**
2. **for all** $a \epsilon$ A **do**
3. $QT(s, a) \leftarrow 0$; $kT(s, a) \leftarrow 0$;
4. **end for**
5. **end for**
6. **while** not stop_condition() **do**
7. **CHOOSE** $s \epsilon$ S, $a \epsilon$ A
8. **Update rule:**
9. $Q_{t+1}(s_t,a_t) \leftarrow Q_t(s_t,a_t)+\alpha[\mathrm{R}(s_t,a_t)+v]$ where,
10. $v \leftarrow \gamma \max Q_t(s_{t+1},a_{t+1})-Q_t(s_t, a_t)$
11. step $\leftarrow$ step + 1;
12. **if** step $< T_x$ **then**
13. **GOTO**{8};
14. **end if**
15. **if** changes are supposed to occur **then**
16. **for** I $\leftarrow$ 1 to #changes **do**
17. **Choose** $s \epsilon$ S
18. $st(s) \leftarrow$ a new status $st \epsilon$ ST;
19. **end for**
20. **Otherwise** continue()
21. **end if**
22. k-NR(T_x, s, a); // *Algorithm 2*
23. **for** $s \epsilon$ S **do**
24. **for** $a \epsilon$ A **do**
25. $QT(s, a) \leftarrow kT(s, a)$
26. **end for**
27. **end for**
28. **end while**
29. **return** (.,.);

generated from an action policy during the learning stage of the Q-Learning. For each environment state, four instances are generated (one for each action) and they represent the values learned by the agent. Table 1 shows some examples of training instances. Each training instance has the following attributes:

1. attributes for the representation of the state in the way of the expected rewards for the actions: north (N), south (S), east (E) and west (W);
2. an action and;
3. reward for this action.

Algorithm 2 shows that the instances are computed to a new table, denoted as kT, which stores the values generated by the k-NR with the k-NN. Such values represent the sum of the rewards received with the interaction with the environment. Rewards are computed using Eq. 6 which calculates the similarity between two training instances $\boldsymbol{s_i}$ and $\boldsymbol{s_m}$.

$$f(\boldsymbol{s_i}, \boldsymbol{s_m}) = \frac{\sum_{x=1}^{x}(s_{i_x} \times s_{m_x})}{\sum_{x=1}^{x} s_{i_x}^2 \times \sum_{x=1}^{x} s_{m_x}^2} \tag{6}$$

Table 1. Training instances.

State (x,y)	Reward (N)	Reward (S)	Reward (E)	Reward (W)	Action Chosen	Reward Action
(2,3)	−0.875	−0.967	0.382	−0.615	(N)	−0.875
(2,3)	−0.875	−0.968	0.382	−0.615	(S)	−0.968
(2,3)	−0.875	−0.968	0.382	−0.615	(W)	0.382
(2,3)	−0.875	−0.968	0.382	−0.615	(E)	−0.615
(1,2)	−0.144	1.655	−0.933	0.350	(N)	−0.144
...	...	...	...	...	...	...

Algorithm 2: k-NR(T_x, s, a).

for all s ϵ S and s $\neq s_{goal}$ **do**
  costQT $\leftarrow$ cost(s, s_{goal}, QT)
  costQ^* $\leftarrow$ cost(s, s_{goal}, Q^*)
  if *costQT*$_s \neq$ *costQ*$*_s$ **then**

$$kT(s,a) \leftarrow \frac{\sum_{i=1}^{k} HQ_i(.,.)}{k} \tag{5}$$

  end if
end for
return $(kT(s,a))$

The cost function (Eq. 7) calculates the cost for an episode (path from a current state s to the state s_{goal} based on the current policy).

$$cost(s, s_{goal}) = \sum_{s \in S}^{s_{goal}} 0.1 + \sum_{s \in S}^{s_{goal}} st(s) \tag{7}$$

Equation 5 used in Algorithm 2 shows how the k-NN algorithm can be used to generate the arrangements of training instances: here, $kT(s,a)$ is the estimated reward value for a given state s and action a, k is the number of nearest neighbors, and $HQ_i(.,.)$ is the i-th existing nearest neighbor in the set of training instances generated from $QT(s,a)$.

Using the k-NR, the values learned by the Q-Learning are stored in the kT table. This contains the best values generated by the Q-Learning and the values that have been estimated by the k-NR.

We have evaluated different ways of generating the arrangements of instances for the k-NN algorithm with the aim of finding the best training sets. First, we used the full arrangement of instances generated throughout runtime. Second, instances generated inside n time windows were selected, where $A_{[T(n)]}$ denotes an arrangement of $T(n)$ windows. In this core, each window generates a new arrangement and previously instances are discarded. We have also evaluated the efficiency rate considering only the arrangement given by the last window $A_{[T(last)]}$. Finally, we have evaluated the efficiency rate of the agent using the last

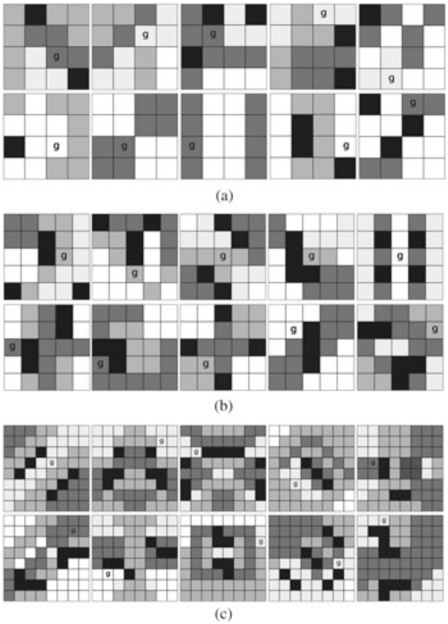


Fig. 2. Simulated environments: (a) 16-state, (b) 25-state, (c) 64-state.

arrangement calculated by the k-NN algorithm - $A_{[T(last),T(k-NN)]}$. The results on these different configurations for generating instances are shown in Sect. 4.

4 Experimental Results

In this section we present the main results obtained from using the k-NR and Q-Learning algorithms. The experiments were carried out in dynamic environments with three different sizes as shown in Fig. 2: 16 (4×4), 25 (5×5) and 64 (8×8) states. Note that a number of states S can generate a long solution space, in which the number of possible policy is $|A||^s|$.

For each environment, ten different configurations were arbitrarily generated to simulate real-world scenarios. The learning process was repeated twenty times for each environment configuration to evaluate the variations that can arise from the agent's actions which are autonomous and stochastic. The results presented in this section for each environment size (16, 25, and 64) are therefore average values over twenty runs. The results do not improve significantly when more scenarios are used (≈2.15 %). The efficiency of the k-NR and Q-Learning algorithms (Y axis in figures) takes into account the number of successful outcomes of a policy in a cycle of steps. We evaluated the agent's behavior in two situations:

1. ♯percent of changes (10, 20, 30) in environment for a window T_x=100. In 64-state environments the changes were inserted after each 1000 steps (T_x=1000)

because in dynamic environments such large environments require many steps to reach a good intermediary policy.

2. ♯percent of changes in environment after the agent finds its best action policy. In this case, we use the full arrangement of learning instances $A_{[T_x]}$, because it gave the best results.

The changes were simulated considering real traffic conditions such as: different levels of traffic jams, partial blocking and free traffic for vehicle flowing. We also allowed for the possibility that unpredictable factors may change traffic behavior, such as accidents, route changing or roadway policy, collisions in traffic lights or intersections, and so on. Changes in the environment were made as follows: for every T_x window, the status of a number of positions is altered random. Equation 8 calculates the number of altered states (♯changes) in T_x.

$$\sharp changes_{(T_x)} = \left(\frac{\sharp states}{100}\right) \times \sharp percent \tag{8}$$

Figure 3 shows the initial experiments with Q-Learning. It is seen that, even with a low change rate in the environment, the agent has trouble converging without the support of exploration strategies. This is because the agent may find a partial policy (a policy found in a cycle of steps before the environment is changed) in response to a certain set of states. To solve this problem, we used the Q-Learning together with the ε-greedy strategy, which allows the agent to explore states with low rewards. With such a strategy, the agent starts to re-explore the states that underwent changes in their status. Figure 4 shows the convergence of the Q-Learning using the ε-greedy strategy in several dynamic environments. More details of the ε-greedy strategy in others scenarios are given in [15]. It can be seen from this experiment that the presence of changed states in an action policy may decrease the agent's convergence significantly. Thus, the reward values that would lead the agent to states with positive rewards can cause the agent to search over states with negative rewards, causing errors. In the next experiments we therefore introduce the k-NR.

4.1 k-NR Evaluation

We used the k-NR to optimize the performance of Q-Learning. The technique was applied only to the environment states where changes occur. Thus, the agent modifies its learning and converges more rapidly to a good action policy. Figure 5 shows how this modification in the heuristic affects convergence of Q-Learning. It is seen that Q-Learning converges slowly without the k-NR (Fig. 4). However using k-NR, the agent rapidly converges to a good policy, because it uses reward values that were not altered when the environment was changed.

It is seen that the proposed approach may accelerate convergence of the RL algorithms, while decreasing the noise rate during the learning process. Moreover, in dynamic environments the aim is to find alternatives which decrease the number of steps that the agent takes until it starts to converge again. The k-NR

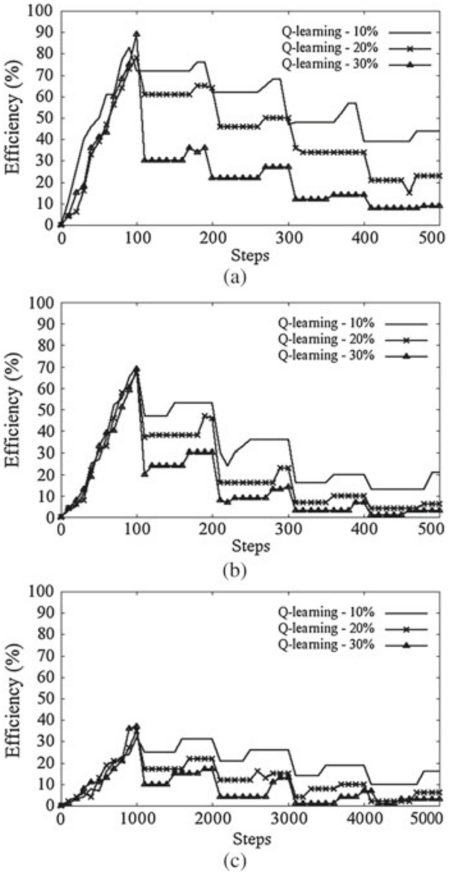


Fig. 3. Performance of the Q-Learning algorithm: (a) 16-state environment, (b) 25-state environment, (c) 64-state environment.

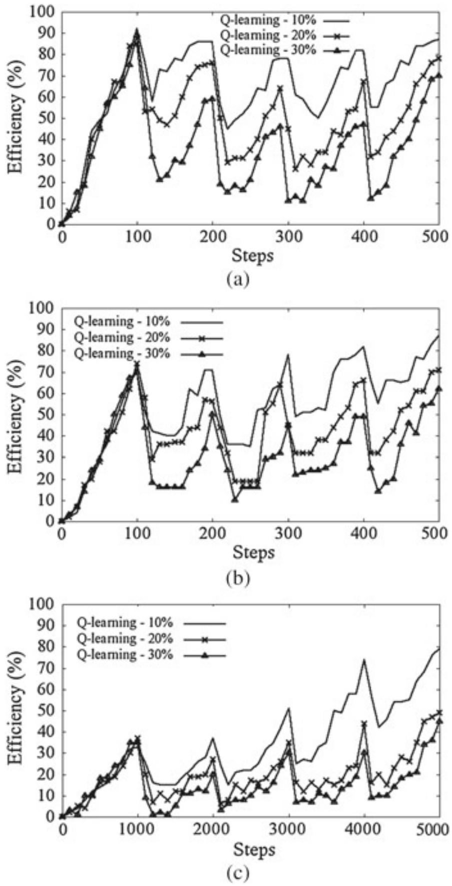


Fig. 4. Performance of the Q-Learning algorithm with ϵ-greedy in a: (a) 16-state environment, (b) 25-state environment, (c) 64-state environment.

algorithm causes the agent to find new action policies, for the states that have had their status altered by the reward values of unaltered neighbor states. In some situations, the agent may continue to convergeeven after a change of the environment. This happens because some states have poor reward values (values that are either too high or too low) as a consequence of too few visits, or too many. Therefore, such states must be altered by giving them more appropriate reward values.

To observe the behavior of the agent in other situations, changes were introduced into the environment only after the agent finds its near-optimal policy (a policy is optimal when the agent knows the best actions). The aim is to analyze the agent's performance when an optimal or near-optimal policy has been

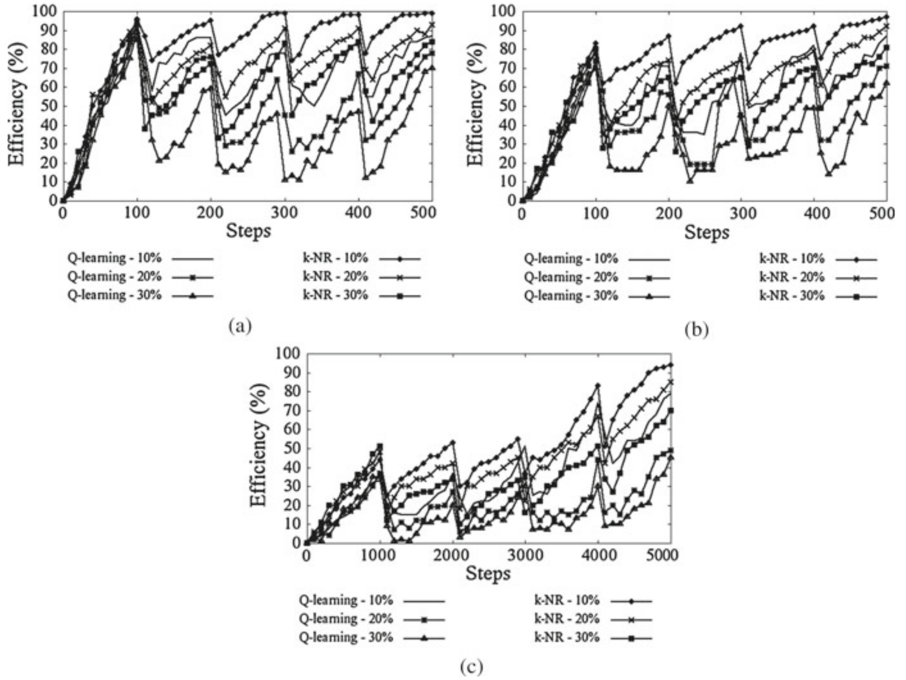


Fig. 5. Performance of the K-NR and Q-Learning algorithms in a: (a) 16-state environment, (b) 25-state environment, (c) 64-state environment.

discovered, and to observe the agent's capacity then to adapt itself to a modified environment.

Enembreck et al. [32] have shown that this is a good way to observe the behavior of an adaptive agent. We have analyzed the agent's adaptation with the k-NR and Q-Learning algorithms. The Q-Learning presents a period of divergence (after some changes were generated), usually a decreasing performance (Fig. 6). However, after a reasonable number of steps, it is seen that there is again convergence to a better policy, as happens when learning begins and performance improves. The decreasing performance occurs because Q-Learning needs to re-explore all the state space, re-visiting states with low rewards to accumulate better values for the future. The ε-greedy strategy helps the agent by introducing random actions so that local maxima are avoided. For example, a *blocked* state that changed to *low jam* must have negative rewards and would no longer be visited.

We used the k-NR algorithm with heuristic to optimize agent performance with the methodology discussed in Sect. 2, which uses instance-based learning in an attempt to solve the problem described in the previous subsection. The heuristic has been applied only to the environment states where changes occur.

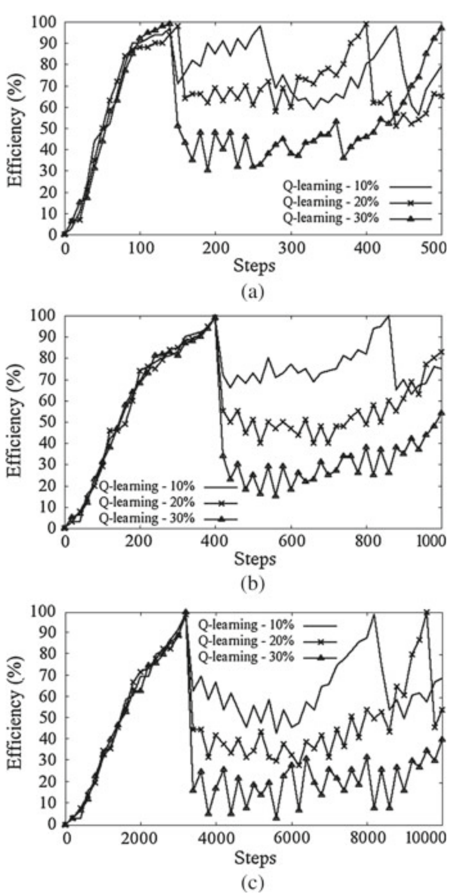


Fig. 6. Agent adaptation using the Q-Learning in a: (a) 16-state environment, (b) 25-state environment, (c) 64-state environment.

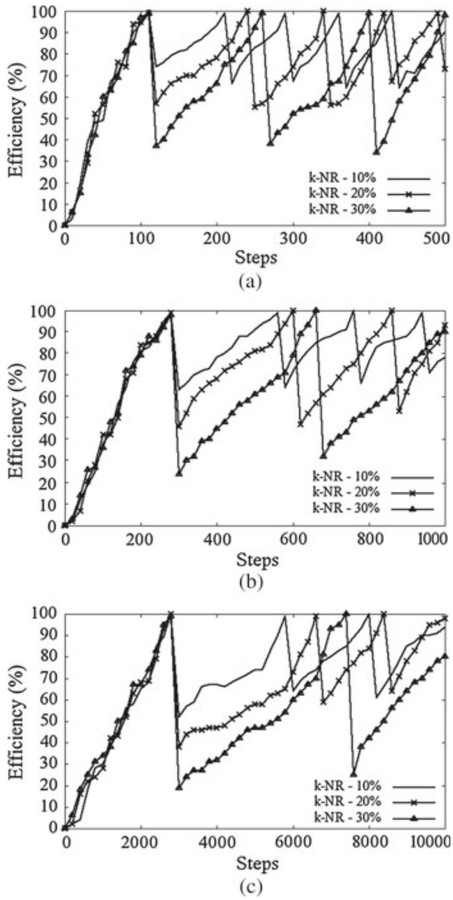


Fig. 7. Agent adaptation using the k-NR in a: (a) 16-state environment, (b) 25-state environment, (c) 64-state environment.

Thus, the heuristic usually caused the agent to modify its learning and converge more rapidly to a good action policy.

Figure 6 also shows that the Q-Learning does not show uniform convergence when compared with k-NR (Fig. 7). This occurs because the k-NR algorithm uses instance-based learning, giving superior performance and speeding up its convergence. The k-NR is able to accelerate the convergence because the states that have had their status altered were estimated from similar situations observed in the past, so that states with similar features have similar rewards.

Table 2 shows the number of steps needed for the agent to refind its best action policy. It is seen that k-NR performs better than standard Q-Learning. In 16-state environments, the agent finds its best policy of actions with 150 steps using the Q-Learning algorithm and 110 steps with k-NR. After changing the

Table 2. Number of steps needed for the agent to find its best policy after changes.

	Before changes		After changes					
			10 %		20 %		30 %	
♯ states	Q	k	Q	k	Q	k	Q	k
16	150	110	130	90	240	140	350	150
25	400	280	440	290	1130	320	2140	380
64	3,430	2,830	5,450	2,950	6,100	3,850	13,900	4,640

environment with 10 %, 20 % and 30 % the k-NR needed 47 % fewer steps on average before it once again finds a policy leading to convergence.

For 25-state environments the agent finds its best action policy in approximately 400 steps using Q-Learning, and in 280 steps with k-NR. In this environment, k-NR uses an average of 30 % fewer steps than Q-Learning, after alteration of the environment. For 64-state environments, the agent needed an average of 3,430 steps to find its best action policy with Q-Learning and 2,830 with k-NR. The k-NR used in average 18 % fewer steps than Q-Learning after environmental change. It is seen that k-NR is more robust in situations where the reward values vary unpredictably. This happens because the k-NN algorithm is less sensitive to noisy data.

5 Discussion and Conclusions

This paper has introduced a technique for speeding up convergence of a policy defined in dynamic environments. This is possible through the use of instance-based learning algorithms. Results obtained when the approach is used show that RL algorithms using instance-based learning can improve their performance in environments with configurations that change. From the experiments, it was concluded that the algorithm is robust in partially-known and complex dynamic environments, and can help to determine optimum actions. Combining algorithms from different paradigms is an interesting approach for the generation of good action policies. Experiments made with the k-NR algorithm show that although computational costs are higher, the results are encouraging because it is able to estimate values and find solutions that support the standard Q-Learning algorithm.

We also observed benefits related to other works using heuristic approaches. For instance, Bianchi et al. [11] proposes a heuristic for RL algorithms that show a significantly better performance (40 %) than the original algorithms. Pegoraro et al. [33] use a strategy that speeds up the convergence of the RL algorithms by 36 %, thus reducing the number of iterations compared with traditional RL algorithms. Although the results obtained with the new technique are satisfactory, additional experiments are needed to answer some questions raised. For example, a multi-agent architecture could be used to explore states placed further from the goal-state and in which the state rewards are smaller. Some of these strategies are found in Ribeiro et al. [34,35]. We also intend to use more than one agent to analyze situations as: (i) sharing with other agents the learning of the best-performing one; (ii) sharing learning values among all the agents simultaneously; (iii) sharing learning values among the best agents only; (iv) sharing learning values only when the agent reaches the goal-state, in which its learning table would be unified with the tables of the others. Another possibility is to evaluate the algorithm in higher-dimension environments, that are also subject to greater variations. These possibilities will be explored in future research.

References

1. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**(3/4), 279–292 (1992)
2. Ribeiro, C.H.C.: A tutorial on reinforcement learning techniques. In: Proceedings of International Joint Conference on Neural Networks, Washington, USA, pp. 59–61 (1999)
3. Tesauro, G.: Temporal difference learning and td-gammon. Commun. ACM **38**(3), 58–68 (1995)
4. Taylor, M., Stone, P.: Using imagery to simplify perceptual abstraction in reinforcement learning agents. J. Mach. Learn. Res. (JMLR) **10**(1), 1633–1685 (2009)
5. Strehl, A.L., Li, L., Littman, M.L.: Reinforcement learning in finite mdps: Pac analysis. J. Mach. Learn. Res. (JMLR) **10**, 2413–2444 (2009)
6. Stula, M., Stipanicev, D., Bodrozic, L.: Intelligent modeling with agent-based fuzzy cognitive map. Int. J. Intell. Syst. **25**(24), 981–1004 (2010)
7. Walsh, T.J., Goschin, S., Littman, M.L.: Integrating sample-based planning and model-based reinforcement learning. In: Proceedings of 14th Conference on Artificial Intelligence (AAAI'10), vol. 1 (2010)
8. Zhang, C., Lesser, V., Abdallah, S.: Self-organization for cordinating decentralized reinforcement learning. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems. AAMAS'10, International Foundation for Autonomous Agents and Multiagent Systems, pp. 739–746 (2010)
9. Wintermute, S.: Using imagery to simplify perceptual abstraction in reinforcement learning agents. In: Proceedings of 24th Conference on Artificial Intelligence (AAAI'10), Atlanta, Georgia, USA, pp. 1567–1573 (2010)
10. Price, B., Boutilier, C.: Accelerating reinforcement learning through implicit imitation. J. Artif. Intell. Res. **19**, 569–629 (2003)
11. Bianchi, R.A.C., Ribeiro, C.H.C., Costa, A.H.R.: Heuristicallyaccelerated Q–learning: A new approach to speed up reinforcementlearning. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS(LNAI), vol. 3171, pp. 245–254. Springer, Heidelberg (2004)
12. Comanici, G., Precup, D.: Optimal policy switching algorithms for reinforcement learning. In: Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10), pp. 709–714 (2010)
13. Banerjee, B., Kraemer, L.: Action discovery for reinforcement learning. In: Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10), pp. 585–1586 (2010)
14. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)

15. Ribeiro, R., Enembreck, F., Koerich, A.L.: A hybrid learningstrategy for discovery of policies of action. In: Sichman, J.S., Coelho, H., Rezende, S.O. (eds.) IBERAMIA-SBIA 2006. LNCS (LNAI),vol. 4140, pp. 268–277. Springer, Heidelberg (2006)
16. Jordan, P.R., Schvartzman, L.J., Wellman, M.P.: Strategy exploration in empirical games. In: Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10),Toronto, Canada, vol. 1, pp. 1131–1138 (2010)
17. Amato, C., Shani, G.: High-level reinforcement learning in strategy games. In: Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10), pp. 75–82 (2010)
18. Spaan, M.T.J., Melo, F.S.: Interaction-driven markov games for decentralized multiagent planning under uncertainty. In: Proceedings of 7th International Conference on AAMAS, Estoril, Portugal, pp. 525–532 (2008)
19. Mohammadian, M.: Multi-agents systems for intelligent control of traffic signals. In: Proceedings of International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce, Sydney, Australia, p. 270 (2006)
20. Le, T., Cai, C.: A new feature for approximate dynamic programming traffic light controller. In: Proceedings of 2th International Workshop on Computational Transportation Science (IWCTS'10), San Jose, CA, USA, pp. 29–34 (2010)
21. Sislak, D., Samek, J., Pechoucek, M.: Decentralized algorithms for collision avoidance in airspace. In: Proceedings of 7th International Conference on AAMAS, Estoril, Portugal, pp. 543–550 (2008)
22. Dimitrakiev, D., Nikolova, N., Tenekedjiev, K.: Simulation and discrete event optimization for automated decisions for in-queue flights. Int. J. Intell. Syst. **25**(28), 460–487 (2010)
23. Firby, R.J.: Adaptive execution in complex dynamic worlds. Ph.D. thesis, Yale University (1989)
24. Pelta, D., Cruz, C., Gonzlez, J.: A study on diversity and cooperation in a multiagent strategy for dynamic optimization problems. Int. J. Intell. Syst. **24**(18), 844–861 (2009)
25. Drummond, C.: Accelerating reinforcement learning by composing solutions of automatically identified subtask. J. Artif. Intell. Res. **16**, 59–104 (2002)
26. Butz, M.: State value learning with an anticipatory learning classifier system in a markov decision process. Technical report, Illinois Genetic Algorithms Laboratory (2002)
27. Koenig, S., Simmons, R.G.: The effect of representation and knowledge on goal-directed exploration with reinforcement learning algorithms. Mach. Learn. **22**(1/3), 227–250 (1996)
28. Bianchi, R.A.C., Ribeiro, C.H.C., Costa, A.H.R.: Accelerating autonomous learning by using heuristic selection of actions. J. Heuristics **14**, 135–168 (2008)
29. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Trans. Pattern Analysis Mach. Intell. **20**(3), 226–239 (1998)
30. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Mach. Learn. **6**(1), 37–66 (1991)
31. Galvn, I., Valls, J., Garca, M., Isasi, P.: A lazy learning approach for building classification models. Int. J. Intell. Syst. **26**(8), 773–786 (2011)
32. Enembreck, F., Avila, B.C., Scalabrini, E.E., Barthes, J.P.A.: Learning drifting negotiations. Appl. Artif. Intell. **21**, 861–881 (2007)

33. Pegoraro, R., Costa, A.H.R., Ribeiro, C.H.C.: Experience generalization for multiagent reinforcement learning. In: Proceedings of XXI International Conference of the Chilean Computer Science Society, Punta Arenas, Chile, pp. 233–239 (2001)
34. Ribeiro, R., Borges, A.P., Enembreck, F.: Interaction models for multiagent reinforcement learning. In: Proceedings of International Conferences on Computational Intelligence for Modelling, Control and Automation; Intelligent Agents, Web Technologies and Internet Commerce; and Innovation in Software Engineering, Vienna, Austria, pp. 464–469 (2008)
35. Ribeiro, R., Borges, A.P., Ronszcka, A.F., Scalabrin, E., Avila, B.C., Enembreck, F.: Combinando modelos de interao para melhorar a coordenao em sistemas multiagente. Revista de Informtica Terica e Aplicada **18**, 133–157 (2011)

Part IV

Information Systems Analysis and Specification

Building Domain-Specific Modeling Languages for Frameworks

Matheus C. Viana(✉), Rosângela A. D. Penteado, and Antônio F. do Prado

Computing Department, Federal University of São Carlos,
Rd. Washignton Luis Km 235, São Carlos, SP, 13565-905, Brazil
{matheus_viana, rosangela, prado}@dc.ufscar.br

Abstract. Frameworks support application development improving productivity and quality. However, due to the number of hot stops to configure them, reusing frameworks may be a complex task which can make developers misuse them and insert defects in the application code. In order to reduce the complexity of this task, we propose increase its abstraction level by using Domain-Specific Modeling Languages (DSML) based on framework features to model and to generate applications that reuse these frameworks. In our approach, framework features and their hot spots are identified from an analysis of its source code and documentation, and then a DSML and a set of templates are constructed. With this DSML, applications reusing the framework can be modeled and generated, protecting developers from framework complexity and decreasing the time spent on the implementation. We illustrate our approach using GRENJ framework as example.

Keywords: Framework · Feature · Domain-specific modeling language · Template · Code generation

1 Introduction

Frameworks are software skeletons that can be instantiated into applications [1]. Their reuse improves productivity, since applications are not developed from scratch. However, framework instantiation is a complex task, due to the internal structure and hot spots that are necessary to know in order to reuse a framework correctly. The incorrect use of frameworks by developers can insert defects in the application code, so that the productivity expected by reusing framework is not achieved. Developers spend a lot of time with maintenance activities to correct theses defects [2,3].

Solutions have been proposed to mitigate framework instantiation complexity, such as manuals, cookbooks and pattern languages. However, these solutions only guide developers and the instantiation of the framework based on the application requirements is still executed by developers and relies on their skills.

In this paper we propose a more effective approach, based on the building of Domain-Specific Modeling Languages (DSML) to facilitate framework instantiation. Our approach is divided in two phases: (1) Domain Engineering, in which

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 191–206, 2013.
DOI: 10.1007/978-3-642-40654-6_12,

a DSML and its code generation mechanism are created based on the framework features and the information necessary to reuse them; (2) Application Engineering, in which this DSML is used to model and to generate applications. A DSML protects developers from framework complexity in two ways: (1) providing higher abstraction elements, framework features, that hide details of the framework internal structure; and (2) configuring the framework hot spots automatically through code generation.

There are also other works which have proposed building DSML for frameworks [4–6]. However, they do not explain clearly how they identify the features of the frameworks. Our approach presents a systematic way to identify the domain features of the framework and the information which is necessary to reuse its classes.

The remainder of this paper is organized as follows: Sections 2 and 3 present, respectively, frameworks and Domain-Specific Modeling Languages; Section 4 shows the approach of framework reuse supported by Domain-Specific Modeling Language; Section 5 presents the use of the proposed approach to build the GRENJ Framework DSML; Section 6 discusses some related works; Section 7 presents the conclusions about this work.

2 Frameworks

Frameworks support software development by providing reuse of code and design. They are composed of two parts: frozen spots, which constitute unchangeable parts, regardless of the applications; and hot spots, which are classes directly reused by the applications. Hot spots can contain hook methods that need to be overridden or invoked in order to customize the framework according to application requirements [7].

A framework is classified according to the way its hot spots are accessed: (1) white box, when such access occurs through inheriting their classes and overriding their methods, (2) black box, when the access occurs through composition, and (3) gray box, when it occurs through the two previous ways. According to their purpose, frameworks can also be classified as: (1) System Infrastructure Frameworks (SIF), which simplify the development of software that controls low-level operations; (2) Middleware Integration Frameworks (MIF), which increase the modularization of applications; (3) Enterprise Application Frameworks (EAF), which are used to instantiate applications for specific domains [2,3].

In this paper we used GRENJ Framework to exemplify our approach. This framework is a white box EAF that addresses the domain of rental, purchase, sale and maintenance transactions of goods or services [8]. The main reasons for choosing GRENJ Framework are: (1) it contains all kind of features, e.g., mandatory, optional and variable; (2) applications within its domain are widely adopted by industry and academia; (3) its code is open; (4) we have developed applications reusing it, so we have full knowledge about its code, hot spots and functionality.

3 Domain-Specific Modeling Languages

Modeling is the act of creating a model to get a better understanding of what is to be built and a more abstract view of a problem solution. It aims at facilitating the comprehension of the solution and serves as documentation [9].

A modeling language is formed by: an abstract syntax, that defines the elements and the rules of the language and usually is represented by metamodel; and a concrete syntax, which defines its notation. The most common notations for modeling languages are textual, graphical and tree-view [10].

In place of general-purpose modeling languages, such as UML, a Domain Specific Modeling Language (DSML) can be used to model applications with elements that are appropriate to the addressed domain. Since there is a greater similarity between the DSML elements and the application features, mapping the application requirements is easier [11] and transformations from application models are more effective. Templates can be used as skeletons to generate several kinds of artifacts [9], such as application code, documentation, test code and other models.

The main advantages provided by the use of DSMLs are: (1) better separation between the logic of the software and the details of the technologies employed in its implementation, (2) greater facility in creating reusable artifacts, and (3) better quality of the software that are produced since their features are well defined and their code can be generated [12].

4 The Proposed Approach

Figure 1 shows the two phases of the proposed approach: Domain Engineering and Application Engineering. Subsections 4.1 and 4.2 present these phases.

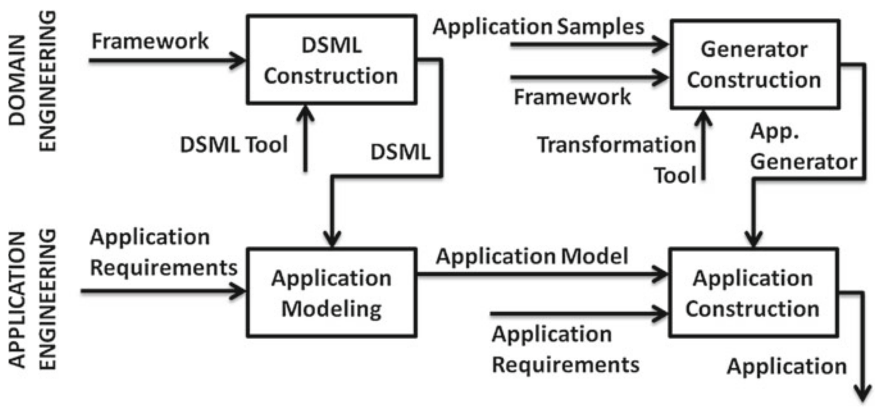


Fig. 1. The proposed approach phases and their activities.

4.1 Domain Engineering

Domain Engineering phase aims at building an environment composed of two parts: a DSML and an application generator. Thus, DSML Construction and Generator Construction activities are carried out to build these parts, as shown in Fig. 1.

DSML Construction activity starts with the creation of DSML metamodel, whose metaclasses represent the framework features. In order to identify these features, framework documentation and code are analyzed through the following steps:

1. Identify for a class that represents a framework feature, which is a class which can be directly extended (white box) or instantiated (black box) by the applications. This class is a hot spot and define a functionality of the framework;
2. Identify the superclasses of the class identified in step 1;
3. Identify the classes targeted by association relationships coming from the classes identified in steps 1 and 2;
4. Search for hook methods in the class identified in step 1 and 2 and identify which information is necessary to invoke/override them;
5. Include a metaclass in the metamodel to represent the class identified in step 1. Also include attributes and/or relationships in this metaclass to store the necessary information to reuse the class it represents;
6. Repeat steps 1 to 5 until all framework features have been identified.

Class models are the first artifacts to be analyzed, because they provide a wide vision of the framework classes. A framework may also be documented by a Pattern Language (PL), which specifies the classes representing the domain features of the framework [13]. The framework code also needs to be analyzed to confirm the classes identified in the documentation, as models may be inconsistent with the current state of the code. Moreover, class models in analysis level contain only the main classes of the framework and the hot spots cannot be identified from these models.

Figure 2 shows part of the GRENJ framework DSML metamodel. Feature, Attribute, Operation and Parameter metaclasses provide common properties to domain-specific features. Applications which reuse GRENJ framework need a class that extends Resource class. This class is a GRENJ Framework hot spot and is concerned with the goods or services involved in transactions. Therefore, Resource metaclass was included in the metamodel to represent it. As a hook method in Resource class requires to know which classes represent the resource types in the applications, a relationship between Resource and ResourceType metaclasses was created to store this information.

The metamodel represents the abstract syntax of the DSML in which the domain features are defined. The concrete syntax, which defines the graphical notation of the DSML, must also be built so that developers can use it to model applications [10]. How the concrete syntax is built depends on the tool used to construct the DSML.

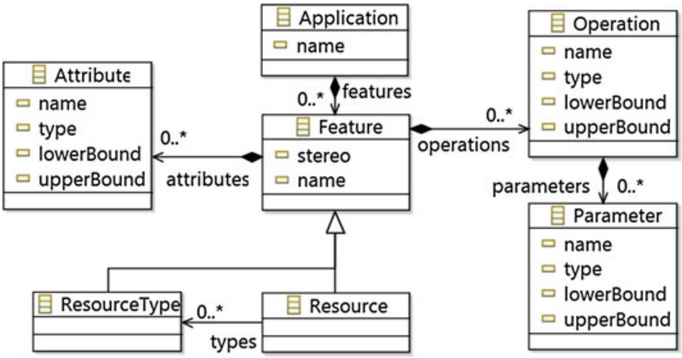


Fig. 2. An example of a framework DSML metamodel.

Generator Construction activity can be carried out building the application generator of the DSML, which is composed of a set of templates. Transformations languages, such as Java Emitter Templates (JET) [9], can be used to implement these templates.

A template is formed by: fixed parts, consisting of all parts that remain the same in applications generated from the template; variable parts, consisting of parts that change from one application to another. Usually, in XML-based template languages, such as JET, the fixed parts consist of texts and the variant parts consist of tags. The best tactic to create templates is to analyze classes of applications that reuse the framework and identify their fixed and variable parts. For example, an application class that extends Resource class of GRENJ Framework should be similar to this:

```
public class Movie extends Resource {
   private int year;
   public Class[] typeClasses() {
      return new Class[] { Category.class, Genre.class };
   }
```

And the JET template to generate Resource subclasses should be:

```
public class <c:get select="$feature/@name"/> extends Resource {
   <c:iterate select="$feature/attributes" var="a">
   private <c:get select="$a/@type"/> <c:get select="$a/@name"/>;
   </c:iterate>
   public Class[] typeClasses() {
      return new Class[] {
      <c:iterate select="$feature/types" var="rt" delimiter=",">
          <c:get select="$rt/@name"/>.class\
      </c:iterate> };
   }
```

In addition to the templates that generates application code, other templates can indicate to the transformation tool to generate the database script, to create the application package, to include a copy of the framework, and so on.

4.2 Application Engineering

Application Engineering phase comprehends Application Modeling and Application Construction activities with the use of the DSML and the application generator constructed in the Domain Engineering phase, as shown in Fig. 1.

In Application Modeling activity, developer uses the DSML to create an application model based on the application requirements. Due its domain-specific aspect, DSML prevents the developers from creating relationships that are not defined in its metamodel and it provides a validation mechanism of the application models, which verifies if all obligatory classes were instantiated among other things.

In Application Construction activity, templates are implemented to generate code and other artifacts from the application model. As code generation is automated, it improves productivity of application development. Although all code responsible to reuse the framework can be generated, it may be necessary to add code because a framework cannot predict all the particularities of the applications, whether using a DSML or not. Generally, the code which is added is responsible for behavioral aspects of the application such as the content of application-specific operations.

Application-specific attributes and methods should be inserted into the application model to avoid inconsistency between the model and the code. Moreover, methods which are added or modified must be protected so that their contents can remain in case of the application code is generated again. Usually, transformation tools provide mechanisms that keep the alterations made by the developer, such as the implementation of the body of a method, even when the application code is regenerated.

5 Example of DSML: GRENJ Framework

We used GRENJ Framework in order to exemplify the use of the approach of framework reuse supported by a DSML. GRENJ Framework DSML was built by using the Graphical Modeling Framework (GMF) and Java Emitter Templates (JET), both available in the Eclipse IDE. Although we have chosen these tools, others could be used as well, such as Generic Modeling Environment (GME) [14] and xPand [9]. The proposed approach is intended to be tool-independent.

This section is composed as follows: Subsection 5.1 presents the GRENJ Framework Domain Engineering; Subsection 5.2 describes Car Rental Application Engineering; Subsection 5.3 shows a study with the development of applications in the domain of rental transactions reusing GRENJ Framework with and without its DSML.

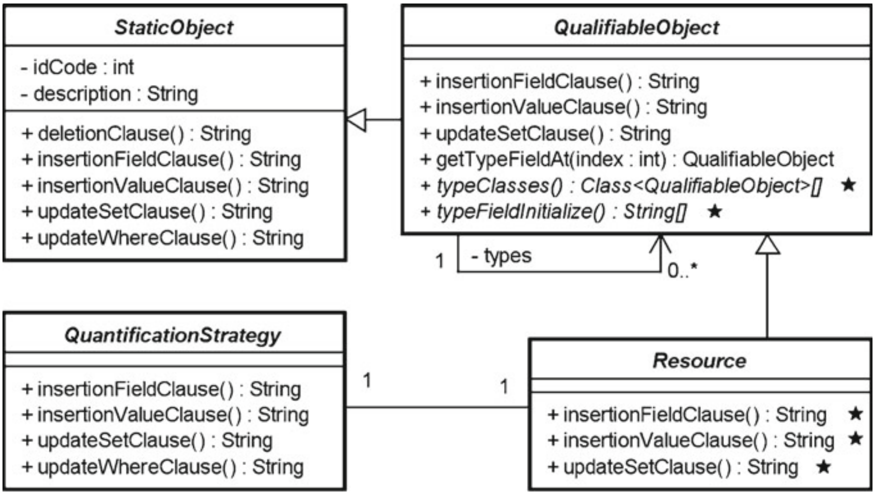


Fig. 3. Classes related to the Resource feature.

5.1 GRENJ Domain Engineering

In DSML Construction activity, GRENJ Framework features were identified by analyzing the classes (hot spots) of this framework that must be extended in the applications, as described in Sect. 4.1. Figure 3 shows a GRENJ Framework model with the classes related to the resource feature.

In Fig. 3 the hook methods that need to be overridden by an application class which extend the Resource class are highlighted by stars. These Hook methods were analyzed in order to identify the information required by the hot spots. For example, it was identified in the code of insertFieldClause hook method that Resource subclasses require the class the implements the resource quantification in the applications.

```
public String insertFieldClause() {
   String fdClause = super.insertFieldClause();
   String qtClause = quantification.insertFieldClause();
   if ( clause != null ) {
     fdClause += ", " + qtClause;
   }
   return fdClause;
}
```

Table 1 shows a list with some of the features identified in GRENJ Framework. These features are the minimum necessary to develop applications that deal with rental transactions reusing the framework GRENJ. Resource type is used to define classes that classify the resources. If the resource type has subtypes, it is a nested type. Otherwise, it is a simple type. Resource quantification identifies whether the resource is unique or can have copies. Resource rental

Table 1. GRENJ Framework features for rental transaction applications.

Feature	Hot spots	Information for Instantiation
All features	-	The name of the class which represents the feature, its attributes, its operations and their parameters.
Resource	Resource	Resource types and quantification.
Resource type	SimpleType, NestedType	-
Resource quantification	SingleResource, ResourceInstance	Is the resource single or instantiable?
Resource rental transaction	ResourceRental	Resource and destination party.
Destination party	DestinationParty	-

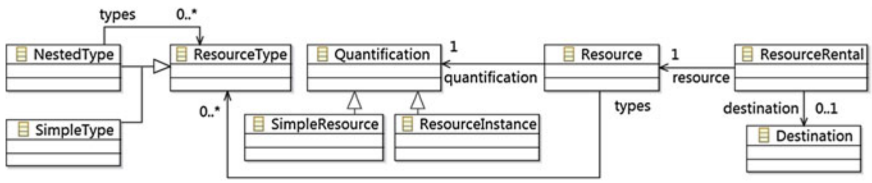


Fig. 4. GRENJ framework DSML metamodel with the features presented in Table 1.

transaction represents the leasing transaction of resources and destination party represents who orders the resource rental.

Figure 4 shows the GRENJ Framework DSML metamodel with the features described in Table 1. All metaclasses that represent domain features extend the Feature metaclass shown in Fig. 2. The multiplicity of the relationships defines how many subclasses of the targeted feature can associate with a subclass of the source feature in the application models.

GRENJ Framework Generator Construction activity was accomplished with use of JET transformation language [9]. This language allows the domain engineer to implement a set of templates and compile it as an Eclipse IDE plug-in that can access the application models and generate code and other artifacts.

A JET template is a XML-format file whose text represents the fixed part of the file that it originates and the variant part is formed by tags which refer to the information to be obtained from the models. Samples of classes of applications that reuse GRENJ Framework were analyzed, such as it was described in Sect. 4.1, so that the fixed and the variant parts of the templates could be identified.

5.2 Car Rental Application Engineering

In order to exemplify the Application Engineering process, the Car Rental Shop application were developed by using the DSML and the templates that were created in the Domain Engineering process described in Sect. 5.1.

The GRENJ Framework DSML and templates originate a set of plug-ins for the Eclipse IDE. The integration of these plug-ins in Eclipse IDE resulted in a CASE tool that supports the developing of applications with the reuse of GRENJ Framework.

In Car Rental Shop Application Modeling activity, the developer creates a model selecting the domain features of GRENJ framework based on the application requirements listed in Table 2. This is similar to create a class model in analysis level.

Figure 5 presents the model of the Car Rental Shop Application which was created with the use of the GRENJ Framework DSML based on the requirements shown in Table 2. In this model, the classes of GRENJ framework that are being

Table 2. Car Rental Shop Application requirements.

#	Description
1	The shop rents cars for customers registered in the application. The attributes of a car are: code, description and number of doors.
2	There can be one or more vehicles for each car registered. A vehicle has license plate, year, color and it can be available or not.
3	A car is classified by a category that determines its rental price.
4	Code, name, address and phone number are the data that need to be registered about the customers.
5	The information to register a car rental is: number, date expected returning date, real returning date, customer, vehicle and total price. Total price is calculated on the number of days between the rental date and real returning date of the vehicle multiplied by the rental price determined by the car category of the vehicle.

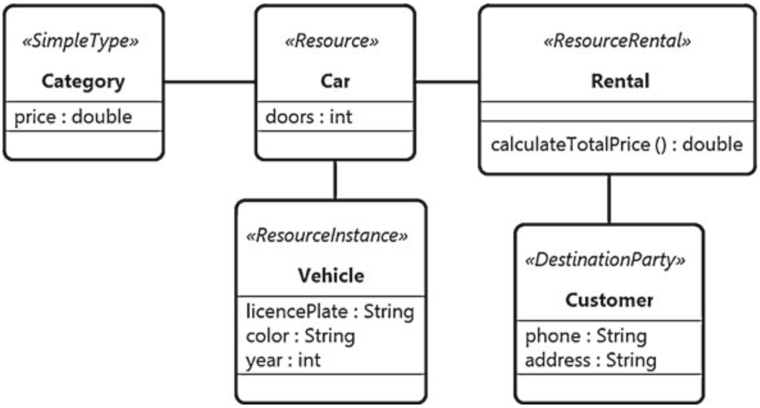


Fig. 5. Car Rental Shop Application model created with GRENJ Framework DSML.

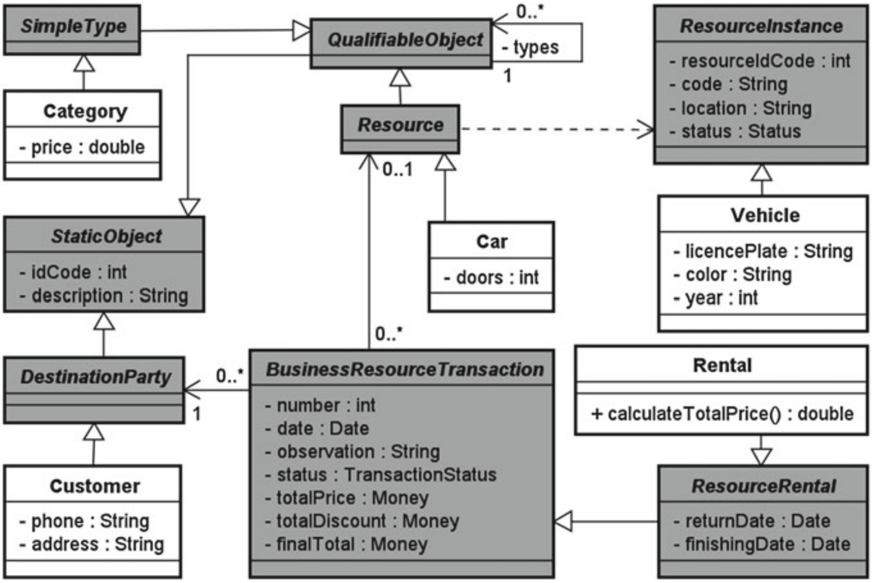


Fig. 6. Car Rental Shop Application class model.

reused by the Car Rental Shop application are identified by the stereotypes and the names of the subclasses are in bold.

Figure 6 illustrates a class model of the Car Rental Shop Application, in which the classes highlighted in grey are from GRENJ Framework. This class model is equivalent to the model created with GRENJ Framework DSML shown in Fig. 5. Most of the attributes specified in the requirements in Table 2 were not added to the classes of the application model shown in Fig. 5 because these attributes are inherited from GRENJ Framework classes and makes no sense to repeat them on the models. Our approach assumes that the application developer knows the framework.

In Application Construction activity, the Car Rental Shop application model was integrated with the templates. Thus the code and all the file structure of the application were generated. The code of the framework was also copied to the application code. A combination of the application model presented in Fig. 5 with the JET template for Resource subclasses, presented in Sect. 4.1, generates the code of the Car class:

```
public class Car extends Resource {

  private int doors;

  public Class[] typeClasses() {
    return new Class[] { Category.class };
  }
```

The total price of the rental in this application is based on the price established by the category multiplied by the number of days the customer rented the car. As the framework cannot predict this calculation, it needs to be manually implemented. The calculateTotalPrice method was generated in the Rental class with an empty body. So it was modified in order to calculate the total price in the instances of the Rental class. This method was marked as "generated NOT" to avoid its code to be erased in case of the application code is generated again. The code of calculateTotalPrice method after it was manually modified is:

```
/**
 * @generated NOT
 */
public double calculateTotalPrice() {
  Category category = (Category) getResource().getTypes().get(0);
  int numDays = Period.numberOfDays(getDate(), getReturningDate());
  return category.getPrice() * numDays;
}
```

5.3 GRENJ DSML Evaluation

A study has been performed to compare the time spent on the development of applications with the reuse of GRENJ Framework. Two approaches were applied in this study: (1) an approach in which application-specific code was manually implemented; and (2) the approach supported by DSML, in which the application-specific code was generated with the use of GRENJ Framework DSML and templates, such as it was described in the Sect. 5.1.

26 Computer Science undergraduate students participated in this study. They were trained to develop applications reusing GRENJ Framework through both approaches in order to get knowledge about the framework and the tools used in each approach. These students had previous knowledge about Java programming and UML modeling.

Two applications involving resource rental transactions were developed by the students: the first one, named Library Application, to check books out from a library; and the second one, named Hotel Application, for checking guests in a hotel. These applications have the same level of complexity of the Car Rental Shop application (Sect. 5.2). The students were divided into two groups, G1 and G2, with 13 students each one. These groups should carry out the following tasks:

- G1 development of the application for the hotel with the reuse of GRENJ Framework through manual programming and development of the application for the library with the reuse of GRENJ Framework supported by its DSML;
- G2 development of the application for the library with the reuse of GRENJ Framework through manual programming and development of the application for the hotel with the reuse of GRENJ Framework supported by its DSML.

To perform the study, the students received a document containing the requirements of the applications they had to develop. Each student also received a class model in analysis level of the application he/she had to develop by manual programming. The applications were developed by using the Eclipse IDE, whose workspace was configured with one project for each application with set of JUnit tests. All computers used by the students had the same configuration of software and hardware.

Each student has developed the applications referent to his/her group individually and measured the time spent in the development of each application. After finishing the development of an application, the students had to pause the chronometer and run its JUnit tests to verify whether or not the application worked correctly. In case of test fail, the students had to write in a form a description of the fail and activate the chronometer again to correct the defects they found. The application was considered finished only after the tests had shown no fail.

To evaluate the use of DSMLs for frameworks, the study worked with two hypotheses:

- Null Hypothesis, H0—the DSML does not make the process of application development with the reuse of frameworks easier and faster; and
- Alternative Hypothesis, H1—the DSML make the process of application development with the reuse of frameworks easier and faster.

The applications developed by the students had approximately 18,400 lines of code, from which 18,000 correspond to GRENJ Framework code and only 400 belong to the code generated or implemented during the application development. Due to idiomatic patterns imposed by GRENJ Framework, the codes of the applications developed by the students are similar, whether they were manually implemented or generated. Moreover, the number of lines of the code generated or implemented during the application development was small due to the reuse of code provided by GRENJ Framework.

Table 3 presents the averages (AVG) and the standard deviation (SD) of the time spent by students of each group (G1 and G2) in the development of the applications. It was possible to observe that there was an average reduction of about 89.3 % for Library Application and 87.5 % for Hotel Application. It results in a general reduction of 88.5 % in the time spent on developing applications with reuse of GRENJ Framework through its DSML when compared to time spent on the manual implementation approach. The times in Table 3 are measured in

Table 3. Average of time spent on the development of the applications.

App.	Manual Programming			DSML		
	Group	AVG	SD	Group	AVG	SD
Library	G2	60’31”	18’59”	G1	6’28”	1’23”
Hotel	G1	44’14”	14’21”	G2	5’32”	0’38”
General Average		52’53”			6’00”	

minutes (‘) and seconds (“) and consider the development of the applications and the successful execution of the application tests.

This study shown that the development of applications with reuse of frameworks spends less time when performed with the use of a DSML than when it is done through manual programming. However, a long time is spent in the execution of the Domain Engineering process in order to construct the DSML and the templates of the framework. Therefore, the approach supported by DSML represents an advantage only when several applications are developed, making the sum of the time spent on Domain Engineering phase and the time spent on the engineering of the applications lesser than the time spent developing these same applications manually. For example, the Domain Engineering process of GRENJ Framework was accomplished in, approximately, 5 h. The sum of 5 h and the times shown in Table 3 demonstrates that, considering GRENJ Framework, our approach supported by DSML becomes worthwhile when 6 or more applications are developed.

Besides the time spent in the development of the applications, the problems shown by the application tests were analyzed in order to identify when the students made more mistakes: with or without the use of the DSML. This analysis was based on the information the students wrote in the form after they had run the tests.

Table 4 shows the number of times which some problems was found in the applications developed by all students in the study. These problems does not include compilation errors, because the applications were tested only when they could be run. The students made more mistakes while reusing the framework without the use of the DSML. This occurred, due to the fact that, without the support of a DSML, there is a higher probability of defect insertions in the code, e.g., mistyping, misuse of the controls, lack of methods required by the framework and so on. In some cases, multiple hot spots of GRENJ framework require the same information, usually, when many classes has the same hook

Table 4. The number of times which some problems was found in the applications developed by the students.

Problem	Number of times it happened	
	Without DSML	With DSML
Missing hook method.	8	0
Hook method implemented incorrectly.	13	0
Different hook methods that require the same information are implemented inconsistently.	6	0
A framework feature was incorrectly selected to implement an application requirement.	2	2
A class attribute previously provided by the framework were included in a class application which extends it.	3	4
Total	32	6

method. Some students implemented these hook methods returning different values, resulting in inconsistencies. It cannot occur with the DSML because it requires each information only once. However, the selection of a framework class to implement an application requirement and the addition of attributes in the application classes rely on the knowledge the developer has about the framework, whether using a DSML or not. This statement could be confirmed, when some students misused framework classes and added redundant attributes, regardless the support of the DSML.

Some restrictions and threats to the validity of this study should be taken into consideration:

– In the development of the applications with the DSML, all students used the DSML and the templates created in the Domain Engineering process described in Sect. 5.1. In this manner, there would be no difference among students on the way that the applications were modeled and the codes were generated.
– It can be argued that the examples developed by the students are simple. However, the applications which were described access most types of GRENJ Framework hot spots. If two or more hot spots request similar information, for example, class names, then these hot spots are the same type.
– The addition of functionality not predicted by GRENJ Framework was not verified in this experiment because this activity can be done only by a manual programming approach.

6 Related Works

In this section, some papers reporting the difficulties during the reuse of frameworks and suggesting possible solutions to this problem are presented and compared with the approach proposed in this paper.

Braga and Masiero (2003) proposed building a wizard for the development of applications reusing an EAF. In this process, the developer fills out the wizard forms according to the application requirements and the wizard generates the application code. It is similar to the approach supported by a DSML. However, it does not generate code directly from the application model.

Oliveira et al. (2007) presented a systematic approach for framework reuse based on the Reuse Description Language (RDL), a language designed by these authors to specify framework instantiation processes. In their approach, RDL is used to register the framework hot spots in an XML-format file and a tool, denominated RDL execution environment, accesses this file for the execution of reuse processes and framework instantiations that lead to domain-specific applications. Therefore, their approach depends on the RDL and the RDL execution environment. The main advantage of our approach over theirs is that our approach does not depend on any specific tool or language so that developers can use their favorite tool to construct the DSML.

Antkiewicz et al. (2009) proposed a method for engineering new framework DSMLs by specializing existing approaches to domain analysis, software development, and quality evaluation of models and languages. Besides their work was focused on the reuse of MIF, it does not explain in details how the features of the framework are identified.

Amatriain and Arumi (2011) proposed an approach in which the construction of a DSML occurs in parallel to the development of its framework through iterative and incremental activities. Our approach differs from theirs, mainly, because our approach can be applied on frameworks that were previously developed by other people.

7 Conclusions

In this paper, an approach to framework reuse supported by Domain-Specific Modeling has been presented. This approach improves the efficiency and the quality on the development of applications in a specific domain, since it generates code from application models and prevents the developer from incorrectly accessing the hot spots of the framework.

The proposed approach requires effort in Domain Engineering phase to build the environment for the development of applications that reuse the framework. However, in Application Engineering phase, the total effort comes down to the effort for modeling the application as well as the effort for the refinement of the code, if it occurs. The time to generate the application code is irrelevant compared to the total effort, because it is generated.

The quality of the DSML depends on the identification of the domain features of the framework in the same way as the quality of an application depends on the identification of its classes. The proposed approach mainly aims to provide a sequence of steps which the developer should follow in order to identify the domain features of the framework, model them and create templates. The use of tools only supports these activities, so that the developer can choose his/her favorite tools for each activity.

The DSML does not eliminate the necessity of the developer to know the framework. He/She has to know the hot spots and the attributes and the operations provided by them in order to add to the classes of the applications only properties that are not provided by the framework or even modify the generated code correctly. The DSML is useful to liberate the developer from implementing the code to reuse the framework, saving time and avoiding some mistakes. Thus, the developer can dedicate more effort to implement the functionality not provided by the framework.

Our work focused on creating DSMLs for frameworks identifying the features and the information necessary to instantiate these frameworks from their hot spots. This is the main idea in our work. Although we have worked with Enterprise Application Frameworks, our approach can also be applied to create DSMLs for System Infrastructure Frameworks or Middleware Integration Frameworks.

In further work, we intend to extend the proposed approach to include the development of the framework from domain features models. We also intend to include in the proposed approach mechanisms to generate behavioral aspects, such as the code of application-specific operations.

Acknowledgements. We would like to thank the students of Computer Science Course of UFSCar for participating in our study. We also thank CAPES for financial support.

References

1. Johnson, R.E.: Frameworks = (Components + Patterns). Commun. ACM **40**, 39–42 (1997)
2. Abi-Antoun, M.: Making frameworks work: a Project retrospective. In: Companion to the 22nd ACM SIGPLAN conference on Object-Oriented Programming Systems and Applications, OOPSLA '07, pp. 1004–1018. ACM, New York (2007)
3. Fayad, M., Schmidt, D.C.: Object-oriented application frameworks. Commun. ACM **40**, 32–38 (1997)
4. Oliveira, T.C., Alencar, P.S.C., Lucena, C.J.P.D., Cowan, D.D.: RDL: a language for framework instantiation representation. J. Syst. Softw. **80**, 1902–1929 (2007)
5. Antkiewicz, M., Czarnecki, K., Stephan, M.: Engineering of framework-specific modeling languages. IEEE Trans. Software Eng. **35**, 795–824 (2009)
6. Amatriain, X., Arumi, P.: Frameworks generate domain-specific languages: a case study in the multimedia domain. IEEE Trans. Software Eng. **37**, 544–558 (2011)
7. Brugali, D., Sycara, K.: Frameworks and pattern languages: an intriguing relationship. ACM Comput. Surv. **32**(1), 2–7 (2000)
8. Durelli, V.H.S., Borges, S.S., Penteado, R.A.D., Viana, M.C.: An iterative reengineering process applying test-driven development and reverse engineering patterns. INFOCOMP J. Comput. Sci. Special Edition(1), 1–8 (2010)
9. Gronback, R.C.: Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit. Addison-Wesley, Upper Saddle River (2009)
10. Cuadrado, J., Molina, J.: A model-based approach to families of embedded domain-specific languages. IEEE Trans. Software Eng. **35**, 825–840 (2009)
11. Turki, S., Soriano, T., Sghaier, A.: An MDA application for a virtual reality environment. In: IEEE International Conference on Industrial Technology, 2004, IEEE ICIT '04, vol. 2, pp. 807–812 (2004)
12. France, R., Rumpe, B.: Model-driven development of complex software: a research roadmap. In: Future of Software Engineering, 2007, FOSE '07, pp. 37–54 (2007)
13. Kirk, D., Roper, M., Wood, M.: Identifying and addressing problems in object-oriented framework reuse. Empirical Softw. Eng. **12**, 243–274 (2007)
14. Institute for Software Integrated Systems: Generic Modeling, Environment (2012)

Locating and Correcting Software Faults in Executable Code Slices via Evolutionary Mutation Testing

Pantelis Stylianos Yiasemis(✉) and Andreas S. Andreou

Department of Computer Engineering and Informatics,
Cyprus University of Technology, 31 Archbishop Kyprianou ave.,
Limassol, Cyprus
pantelis.yiasemis@cut.ac.cy, andreas.andreou@cut.ac.cy

Abstract. Software testing is an important phase of software development that helps eliminating the possibility of project failure. As software systems get more complicated and larger in size, testing needs to constantly evolve and provide more "sophisticated" techniques, like automatic, self-adaptive mutation testing, targeting at improving the efficiency and effectiveness of the testing phase by handling the increased complexity that leads to increased demands in time and effort. Mutation testing is the procedure of applying a series of operators on correctly functioning programs so as to induce "faults" that correspond to real, common programming errors and then assess the ability of a set of test cases to reveal those errors. We introduce a novel approach for identifying and correcting faults in Java source code with the use of code slicing, mutation testing and Genetic Algorithms. Three different categories of experiments are used to assess the effectiveness of the proposed solution, demonstrating its applicability on a variety of programs and type of errors. The results are quite encouraging suggesting that the approach is able to dynamically detect faults and propose the appropriate corrections.

Keywords: Mutation testing · Fault localization and correction · Genetic algorithms

1 Introduction

In the rapidly changing reality of software development, software systems become larger and more complicated, offering more functionality and satisfying a richer set of constraints; but usually these come at a cost for quality. A significant research stream in Software Engineering focuses on how to increase the productivity and quality of the produced software. One of the major reasons why the aforementioned issues are hard to tackle is the presence of faults in code. A fault can be an incorrect step, procedure or data definition in a program. Often they are called problems, errors, anomalies, inconsistencies or bugs [1]. Faults in a system lead to failure, which is described as the inability of a system or component to correctly execute the functions defined by the

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 207–227, 2013.
DOI: 10.1007/978-3-642-40654-6_13,

required specifications, as stated in the IEEE standard Glossary of Software Engineering Terminology.

Examples of significant failures include Disney's Lion King in 1994–1995, Intel's Pentium Floating-Point Division Bug in 1994, NASA's Mars Polar Lander in 1999, Patriot Missile Defence System in 1991, Y2K (Year 2000) Bug originated decades back—circa 1974 and the Dangerous Viewing Ahead in 2004 [1].

The software testing process consists of two sub-processes; failure detection that is performed during the execution of a program and debugging, where faults leading to failure are being identified and corrected. Testing is one of the most hard, demanding and time consuming processes of the development cycle; software firms sometimes spend around 50–80 % of the total development time trying to reduce the number of faults in source code. The process of actually detecting faults in a software system takes up to 95 % of the whole code debugging sub-process [2]. Having this in mind, one can safely conclude that there is a need for developing highly efficient and effective fault detection methods in order to assist and improve the code testing process. Succeeding in this endeavor will reduce the time and effort spent on code testing, freeing resources and improving the quality of the produced system, while at the same time increasing the productivity of software developers.

This study aims at introducing an innovative method for code testing in Java by making use of Code Slicing (CS), Mutation Testing (MT) and Genetic Algorithms (GA). More specifically, the benefits of utilizing static CS will be exploited so as to create smaller executable parts of erroneously behaving programs and then change the execution flow by replacing (correcting) statements that contain one or more faults. The problem of actually detecting a fault and proposing the correct replacement for its elimination is quite complicated as it requires searching a large space of candidate solutions; the latter may be successfully tackled with the use of GAs which reduce it to a search optimization problem.

The rest of the chapter is organized as follows: Section 2 presents a short literature overview on fault localization techniques, mutation testing and some Computational Intelligence techniques that are utilized in this area. The next section, describes the technical background of the algorithms used in the proposed solution. Also, a description is given of how mutation testing is performed, along with some information about the mutation operators and a brief introduction to the Kaveri\Indus tool. Section 4 describes the proposed method and how different algorithms are implemented and integrated in a dedicated supporting tool. Section 5 presents the details of the application prototype and the series of the experiments conducted. In Sect. 6 we discuss some limitations of the proposed approach, while Sect. 7 presents our conclusions and outlines future research plans.

2 Literature Overview

As mentioned earlier, code testing is a very time and effort consuming process leading to the conclusion that there is a strong need to achieve high efficiency and effectiveness while at the same time reduce the cost associated with the completion of its tasks. In this context a considerable amount of studies regarding the development of automated testing methods and tools are reported in literature, which introduce a variety of methods and algorithms to limit the human effort required for testing, taking advantage of the continuous advancements in computer processing power. A number of different approaches have been suggested in literature, like Delta debugging, variations of dynamic programming, failure inducing chops and predicate switching. A description of a few such methods is given below, most of them lying in the area of fault localization and/or program slicing. Program slicing constitutes a very promising approach that may assist in the identification and isolation of faulty statements in code.

Program slicing was introduced in 1979 by Weiser [3], while different types of slicing were proposed since then. Static slicing, proposed by Weiser, requires that only information being derived from the program during compilation is used; no information regarding input values or the way the program functions at runtime is utilized. Dynamic slicing, on the contrary, makes use of information that is available during execution of a program with specific input values. In both types of slicing, the slice is created either in a backward or forward manner. Backward slicing is defined by the program statements that have an effect on the slicing criterion, without meaning that all statements that are included in the final slice have to occur before the execution of the statement of the criterion. On the other hand, forward slicing involves all of the statements that are being affected by the statement set as the slicing criterion [4].

Fault Localization (FL) is the process of identifying parts of the code that may contain faults and then examine them to prove that actually faults are present (and where) in those parts. It is a heuristic method based on dataflow tests that supports execution slicing and dicing based on test cases [5]. Agrawal and Horgan [6] observed that when a test case fails during execution then the fault is contained in that slice or that the fault maybe caused by the absence of a statement from that slice thus evaluating wrongly the test case. The slice can then be isolated from the rest of the program and the search for the fault can concentrate on that specific code part. The results of this method showed that some faults inserted by independent observers were detected, but not all of them were included in the produced slices. Black et al. [7] studied the characteristics of a program's slices in an attempt to identify those components that could contain faults. A slicing profile was formed with slicing metrics and dependence clusters to investigate if a reliable tool could be developed so as to identify the more fault-prone components, yielding encouraging preliminary results.

Mutation Testing (MT) is a technique that is based on the insertion of faults in the source code; it was first introduced in late 70s by Hamlet [8] and Demillo et al. [9]. Currently, it is being used in various research studies involving software testing [10, 11]. MT is based on the idea that faults programmers are prone to, are being infused to the initial program in order to create a faulty version called "mutant" which contains a specific change. A set of test case data is used to execute the mutants, and the quality of the said set is assessed by the number of faults it was able to identify.

Genetic Algorithms (GA) are search-based algorithms that rely on the principles of natural selection and genetic reproduction [11, 12]. They constitute a special class of optimization techniques that maintain a population of individuals, each representing a possible solution to the problem in hand. A repetitive evolution of the population takes place, during which genetic operators that mutate and cross-over the individuals are applied trying to reach to better possible solutions in each generation. A fitness function is used to evaluate the fitness of each generation. Both program slicing and GA have been extensively reported in literature; nevertheless, to the best of our knowledge, they we have not been combined yet for fault localization and correction. This gap motivated the present work to exploit the benefits of these two techniques in a single proposed engine. Furthermore, the work of Jiang et al. [13] which describes how software faults may be localized based on a set of testing requirements and program slicing, inspired the encoding scheme of the GA population used. More specifically, Jiang and his colleagues proposed an approach that can identify dependence structures in a program which searches a superset of all possible slices to identify the set of slices that achieves maximum coverage. The framework yielded successful results demonstrating in practice that it is possible to express problems of dependency analysis as search problems and solve them within an acceptable time frame.

Arcuri and Yao [14] proposed a framework for automatic software bug fixing using co-evolution, where both programs and test cases co-evolve, aiming at fixing bugs by influencing each other. The framework requires that both the faulty program and its formal specification are used as inputs. The work included some preliminary experiments that supported its potential applicability for any implementable program; nevertheless, it may prove costly and not efficient for large scale programs as it attempts to evolve the whole program tree. Also, formal specifications are not always available, a fact that limits further the applicability of the framework in such cases.

Genetic Programming (GP) and program analysis were used by Weimer et al. [15] in order to repair off-the-shelf legacy systems. The GP uses the source code to be repaired, a set of negative test cases that exercise the fault and a number of positive test cases that correspond to correct executions of the program. It then modifies the program creating a candidate in a way that the negative test cases do not fail while the positive still pass. The first candidate that passes all test cases is called a primary repair, which is then treated by a program analysis process so as to reduce it to a minimized repair. Program analysis is being used in order to get rid of the irrelevant changes that the algorithm might have created in the rest of the code. As off-the-shelf systems were used the bugs that already existed in the systems were “real” faults made by programmers and were not manually injected in code. This provided better conclusions and the study was the first to work on real programs with real faults. This approach was also adopted in the present study by making use of faulty and successful test cases to guide our GA in evolving its population so as to provide the correct repair for a fault.

3 Technical Background

3.1 Mutation Testing (MT)

MT introduces faults to program code similar to common errors made by programmers. As Hamlet [8] stated, MT assists in finding a number of test cases that can be used to identify real programming faults. The number of possible faults is quite large and so traditional MT targets only faults that are closer to the original code. This theory is based on two hypotheses: Competent Programmer Hypothesis (CPH) and Coupling Effect (CE). CPH states that programmers tend to write nearly correct code, while CE states that test data used to identify simple faults is sensitive enough to identify complex errors as well [16]. Recent studies in literature that deal with high order mutations exist, like Harman et al. [10] and Fraser and Zeller [17]. In this chapter we choose to focus only on first order mutants as these, based on the theory of CE, may be considered good enough for performing adequate testing of program code: Simple mutations created with syntactic changes may represent simple faults, while complex faults are being represented with more complex mutations consisting of more than a single change in the code.

Traditional MT states that we get a set of faulty programs (mutants) from a program P after the application of some syntactic changes on the original code. The transformation rule that is applied in each case producing a mutant is called a Mutation or Mutant Operator (MO). The MOs are designed to modify expressions and variables found in code by insertion, replacement or deletion [18]. To start the mutation analysis, first the original program is executed against a set of test cases to check if it is executed correctly and produces the desired results. Then the set of test cases is run against all the mutants produced by the mutation engine. If the results of the execution for a given mutant are different compared to those of the original program then we may say that the mutant has been "killed", otherwise, in the case in which the results remain the same, that it has survived. Furthermore, after the execution of all the test cases some of the mutants may still survive, so additional runs with new test data should be executed in order to "kill" the surviving ones as well. Finally, if some mutants survive even after employing this additional test data and still return the same results with the original code, then we have the phenomenon of the so-called Equivalent Mutants, which, although syntactically different from the original program, provide the same functionality.

The purpose of testing through mutation methods is usually completed with the calculation of an adequacy score known as Mutation Score (MS). MS defines the quality of a given set of test cases given as input to a program. It is the analogy of the number of "killed" mutants against the number of the non-equivalent mutants. The purpose of mutation analysis is to increase the MS and bring it closer to 1, meaning that the set of test cases used is capable to detect all the faults present in the mutants.

3.1.1 Mujava and Mutation Operators

A variety of different tools for MT have been suggested, dating as back as 1991. Mothra [19] is the first full MT software tool providing a versatile environment for FORTRAN. For Java there exists a tool called Jester, which is basically a simple open source tool for Java code MT [20], which is integrated with JUnit, a well-known code testing environment. MT in Jester does not use any sophisticated algorithm for accelerating the mutation operation, something which results in slow performance, while it does not provide a large number of supported MOs. These two facts make the use of Jester ineffective and non-practical for large sized, real-world programs. MuJava [21] is another Java tool that provides mutation analysis, aiming to study MO related to object oriented programming languages and their features. It is regarded as one of the most complete, in terms of MO supported, and full of features tools for mutation analysis for the Java language. It offers mutations for traditional testing, but also for testing at the class level, by combining two basic technologies, Mutant Schemata Generation (MSG) and byte-code translation. Using MSG it creates “meta-mutants” of the program at the source code level that integrate a number of mutations. Working directly on the byte-code means that only two compilations are needed, the one of the original program and that of the meta-mutants that MSG created. This improves the performance of the tool over other mutation testing tools that compile all the mutants. Based on the above useful characteristics, we decided to utilize MuJava in our study.

MuJava uses two kinds of MO, method-level [22] and class-level operators [23]. Method-level operators change the source code by replacing, deleting and inserting primitive operators. There are six different primitive operators: arithmetic, relational, conditional, shift, logical and assignment operators. Some of these operators consist of a number of other sub-operators (e.g. binary arithmetic and shortcuts).

The class mutation operators in MuJava are categorized in four different sets based on the characteristics of the programming language they affect. These are Encapsulation, Inheritance, Polymorphism and Java-Specific features.

3.2 Indus Java Program Slicer - Kaveri

Program slicing is used to identify portions of the code that either influence or are being influenced by a specific part of program code, the slicing criterion. A large number of studies have been devoted to program slicing but not so many tools have been developed for slicing. Furthermore, most of the existing tools target the C programming language, while languages like C++ and Java are not so popular with these tools.

In Java one may find a limited number of tools that support code slicing, two of which are Jslice and Indus. VALSOFT/Joana tools is another option but it comes with a disclaimer that the tool is essentially the result of work-in-progress and the results are likely to contain faults. As J.P. de Jong [24] explains, sometimes the results may not be the expected ones. Previous work on the topic [25] used JSlice to perform dynamic slicing based on specific input values of the program. In this work we decided to extend the aforementioned work and utilize static slicing. This enhancement was performed in order to produce an executable slice of the original program which would contain all the lines of code affecting the slicing criterion and not only the lines of code that are reached under a specific input value like in dynamic slicing. The executable slice would then give us the ability to fully automate the process of testing our code via the proposed engine. In addition, our choice does not affect the effectiveness of the approach as on one hand the static slice contains all the information needed by the engine to locate and correct the fault, in fact the static slice is a superset of the dynamic, and on the other the cost in time for handling larger slices is negligible. The tool selected was Kaveri/Indus slicer (http://indus.projects.cis.ksu.edu/), which provides a program slicing library that can handle almost all features of Java except dynamic class loading, reflection and native methods. It can produce both backward and forward slices and provides a number of options and features for the application of both kinds of slices.

Kaveri is an Eclipse plugin that provides the program slicing functionality of the Indus slicing tool in the Eclipse platform. It adds toolbars, views and options to the Eclipse IDE, in order to aid program comprehension and help users visualize program slice and other information that may be required for the analysis. Also, it provides the ability to slice Java programs by choosing slicing criteria through the provided GUI, view the slice produced in the Java editor as highlighted statements, perform context-sensitive slicing and execute additive slicing. The selected tool has been applied to numerous Java programs by its developers, the size of which sometimes reached 10,000 lines of code, thus demonstrating its efficiency and applicability.

4 Automatic, Evolutionary Mutation Testing

The proposed approach introduces an efficient, automatic way to define the specific line or the smaller possible set of lines responsible for a fault present in a Java program. Moreover, the approach aims at providing the necessary correction(s) in order to remove the fault. As the number of possible corrections that can be performed to the set of statements of the original code can be quite large, thus making their manual processing hard and time consuming, we slice the original code before applying the mutation testing techniques so as to limit the number of the mutants created.

The use of static backward slicing enables isolating those lines of a program that affect a variable at a given point of interest for any input of the program and at the same time ensures that the slice produced is executable; the latter property is of great importance to achieve an automatic testing process. The slicing criterion is manually set in the desired normalized source code file within the Eclipse environment. The options for the slicing algorithm can then be set through Kaveri's GUI, followed by invocation of Indus to slice the program. The proposed approach combines backward static slicing with mutation testing and improves the efficiency of the fault correction process by utilizing genetic algorithms to evolve mutant solutions that essentially suggest the proper corrections (see Fig. 1).

We represent each possible solution (correction) to our problem as a chromosome of size N, where N is the number of lines contained in the produced slice. Each gene of the chromosome represents a line of the slice and can take any value in the domain [$0, K$], where any value other than 0 represents a specific mutation operator applied on that line and 0 corresponding to the case where no MO has been applied on the specific line. K is the maximum number of supported mutation operators.

Based on this encoding scheme we can extract the total size of the search space for a static backward slice of size N using the following formula:

$$(A_1 + 1)^* (A_2 + 1)^* (A_3 + 1)^* \ldots{}^* (A_N + 1) \quad (1)$$

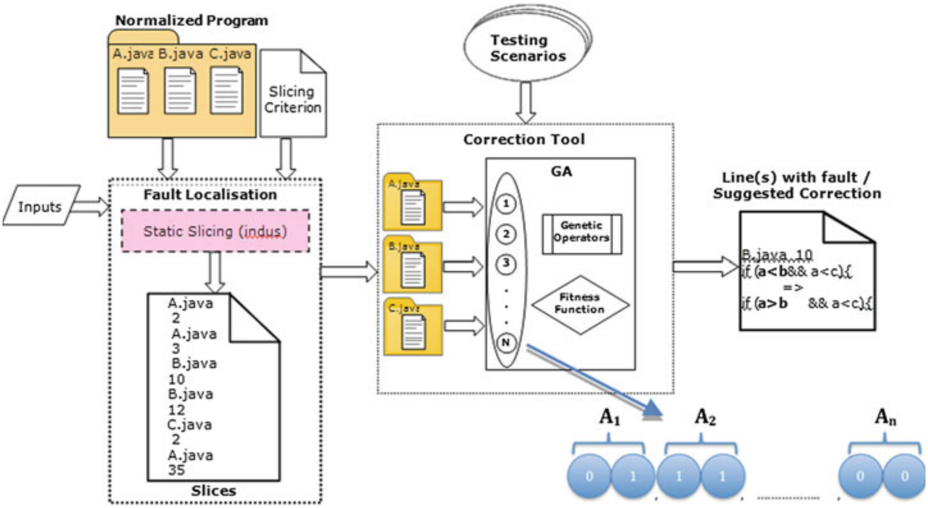


Fig. 1. The proposed hybrid approach for fault localization and correction combining dynamic slicing, mutation testing and Genetic Algorithms.

where A_x is the number of replacements for line x in the slice. As the case of no replacements is valid as well, it is necessary to add 1 to each number of replacements for each line so that the minimum size of the search space is 1 (i.e. no mutations are applied on any line).

Specifically, in Fig. 2 we can observe how a slice with its correction is represented in the genetic algorithm. In the example given 12 lines of code exist in the slice (A_1 – A_{12}), each taking a value from 0 to 15, as we choose, for simplicity's sake, to represent each MO with a decimal gene representing 1 of 15 MOs. In reality the program represents each MO in binary form, on which it applies the different evolutionary operators (cross-over, mutation) to evolve individuals to the next generation.

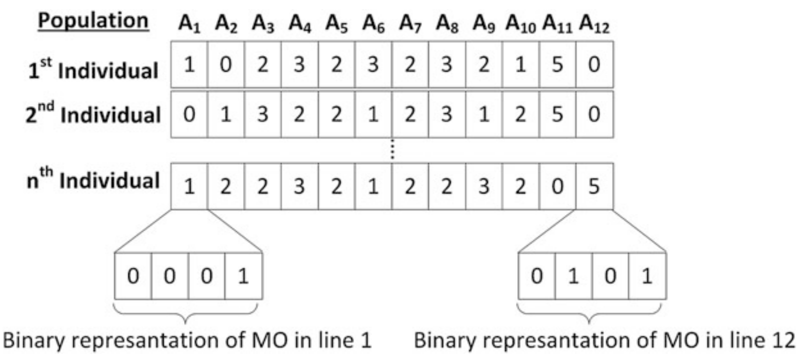


Fig. 2. Analytic representation of the chromosomes in the population.

In the last (n^{th}) individual in Fig. 2 we observe that in the first line the first MO is applied, while in the last line of the slice the fifth MO is applied.

To drive the algorithm to the best possible solution we use a Fitness Function that takes into account the results of the execution of each of the mutated programs, against a number of predefined successful and faulty test cases, that is, test scenarios that execute correctly or not, respectively, on the original unmodified program. Then the algorithm assesses a specific replacement based on two elements: (a) The number of successful test cases that remain successful after the replacement, (b) The number of faulty test cases that become successful after the replacement has taken place. The probability of a line to include a fault increases proportionally to the fitness of its "best" replacement. A specific solution suggests one or more lines that contain a fault. The number of lines contained in each solution affects the fitness of that specific solution. Specifically, each suggested solution (chromosome) is evaluated using the following formula:

$$\frac{\sum_{n=1}^{L}\left(SS_n \cdot SW_n \sum_{j=1}^{S}\left(SSC_{j,n}\right) + FS_n \cdot FW_n \sum_{k=1}^{F}\left(FSC_{k,n}\right)\right)}{N \cdot SLW} \tag{2}$$

where:

SS_n and FS_n are the numbers of successful test case scenarios that remained successful, and of faulty scenarios that turned to successful respectively after replacement n, SW_n and FW_n are weights defining the significance of the successful and faulty test case scenarios respectively, $SSC_{j,n}$ is a constant score for a specific successful test case scenario j after replacement n (in case we want to give a specific successful scenario higher importance over the others), $FSC_{j,n}$ is a constant score for a specific faulty test case scenario j after replacement n (in case we want to give a specific faulty scenario higher importance over the others), SLW is a weight that reflects the importance of the slice size, L is the number of lines contained in the proposed solution, i.e. the number of genes that were graded with a value different of 0, S and F are the number of successful and faulty test scenarios respectively, and N the slice size.

For the algorithm to terminate at least one of the following three criteria has to be met: (i) The predefined maximum number of generations has been reached, (ii) A chromosome has been evolved that yields the highest possible fitness score as expressed by Eq. (2)–this is achieved when the chromosome involves only one line that contains the fault and the proposed replacement converts all faulty test cases to successful, while at the same time successful test case remain successful, (iii) For the last M generations the fitness of the best chromosome becomes equal or lower compared to that of the previous generation and M exceeds the 25 % of the total number of generations set.

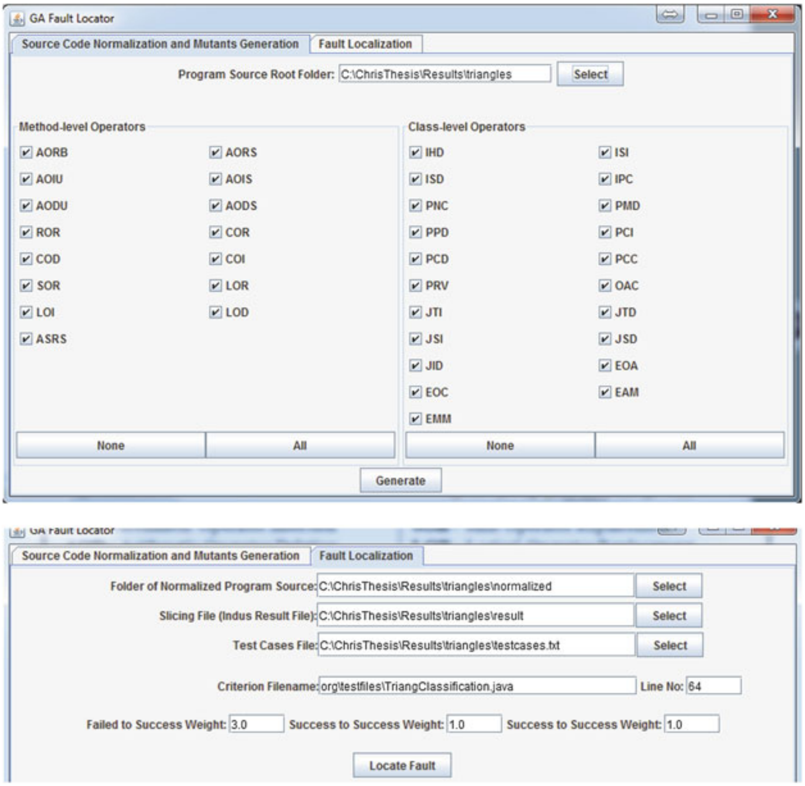


Fig. 3. Screenshots of the supporting software tool.

4.1 Application Issues

A dedicated tool was implemented in Java to support the proposed approach. (Screenshots are provided in Fig. 3). The user is able to select from two tabs, one for the normalization and mutation of the original program's code and one for the genetic algorithm application. First, the user defines, using the left tab (***Source Code Normalization and Mutants Generation*** – Fig. 3) the folder that contains the source files and packages of the Java code to test, as well as the mutation operators to be applied. The available operators to choose from are part of those provided by the MuJava tool; specifically those that fit the purposes of this work (see Table 1).

Upon definition of the aforementioned parameters, the tool first creates a normalized version of the code (removes spaces, comments etc.), then it applies the selected mutation operators on the specified program and finally it creates a new copy (mutant), based on the normalized version, for each mutation. When all mutants are created, the user may move to the second tab (***Fault Localization*** **–** Fig. 3) where the slicing criterion has to be defined according to which the static slice will be created. This is basically the line where a "faulty" result or an erroneous value of a certain variable is observed. At present, the static slice of the program is produced through the

Table 1. Mutation Operators used in our algorithm.

Method Level Mutation Operators	
AOR - Arithmetic Operator Replacement	COD - Conditional Operator Deletion
AOI - Arithmetic Operator Insertion	SOR - Shift Operator Replacement
AOD - Arithmetic Operator Deletion	LOR - Logical Operator Replacement
ROR - Relational Operator Replacement	LOI - Logical Operator Insertion
COR - Conditional Operator Replacement	LOD - Logical Operator Deletion
COI - Conditional Operator Insertion	
Class Level Mutation Operators	
IHD - Hiding variable deletion	PRV - Reference assignment with other comparable variable
IOP - Overriding method calling position change	OAC - Arguments of overloading method call change
ISI - super keyword insertion	JTI - this keyword insertion
ISD - super keyword deletion	JTD - this keyword deletion
IPC - Explicit call to a parent's constructor deletion	JSI - static modifier insertion
PNC - new method call with child class type	JSD - static modifier deletion
PMD - Member variable declaration with parent class type	JID - Member variable initialization deletion
PPD - Parameter variable declaration with child class type	EOA - Reference assignment and content assignment replacement
PCI - Type cast operator insertion	EOC - Reference comparison and content comparison replacement
Polymorphism PCC Cast type change	EAM - Acessor method change
PCD - Type cast operator deletion	EMM - Modifier method change
PCC - Cast type change	

use of the Kaveri/Indus tool within the Eclipse framework, whereas in the future the Eclipse will be replaced by a direct call to the slicing tool of Indus to fully automate the process. The user has to define the test cases file that contain the successful and faulty testing scenarios to be used, as well as the weights that involve three decimal numbers defining the grading weights for the mutations. Finally, the user clicks the ***Locate Fault*** button for the GA to start evolving the mutants. During this process, the user first defines the initial values for the parameters of the Genetic Algorithm, with the domain value for each gene being based on the maximum allowable number of mutations for the line it represents. The results are then stored in a log file that contains the program slice, the fitness function score in each generation and the resulting slice with the correction to the fault, and are presented to the user.

5 Experimental Results

To assess the effectiveness and efficiency of the proposed approach we executed three series of experiments the results of which are reported in this section: Category A includes mutant programs with errors at the level of methods, while category B involves errors induced at the level of classes. Category C evaluates the proposed approach on real-world programs. In all experimental categories, special attention is given on scalability issues, the latter being measured in terms of size, i.e. lines of code of the source code and of the initial slice produced, as well as complexity, i.e. the number of different source files involved that construct the executable slice and the size of the test cases set needed for describing the correct functionality of each program used. The experiments were performed on an Intel i7-2600 processor at 3.4 GHz with support for 3.8 GHz when Intel® Turbo Boost Technology is used and 4 GB of RAM.

The general parameters of the GA's used in the fitness function were as follows: For every slice, the weight of significance for faulty test case scenarios (*FW*) was set to 3.0, while for successful (*SW)* was set to 1.0. The weight that reflects the significance of the slice size (*SLW*) was always equal to 1.0. The number of generations was set equal to 200, thus calculating *M* to 50 (25 % of the number of generations). The specific GA settings were as follows: mutation and crossover probabilities were set to 0.01 and 0.3 respectively, while roulette wheel was used as the selection mechanism. Finally, equal importance was given to all scenarios, with both the constant score for successful and faulty test case scenarios after a replacement (*SSC* and *FSC* respectively*)* being set to 10 for every scenario. The experimental results reported are the average of 25 GA runs.

5.1 Experiment Series A'

This group corresponds to mutations performed at the method level. More specifically, single errors producing first order mutations have been induced for the following operator categories: (i) Arithmetic, (ii) Relational, (iii) Conditional, (iv) Logical and (v) Assignment. The latter was performed via a shift operation thus covering also this specific category of mutation operators.

The sample programs used in the experiments correspond to programming solutions to well-known problems that are usually treated as benchmarks; these programs

Table 2. Results obtained using mutations performed at the method level.

Sample Program	Lines of Source Code	Slice Size	Files	Initial Statement VS *Mutated Statement*	Testing Scenarios	Time (sec)
Credit Card Validation MO: AOD LOC: 57	183	18	3	cachedLastFind=i <vs> *cachedLastFind=i++*	2 Success 6 Fail	23
Triangle Classification MO: ROR LOC: 30	80	41	1	if ((i + j <= k) || (j + k <= i) || (i + k <= j)) <vs> *if ((i + j == k) || (j + k <= i) || (i + k <= j))*	23 Success 11 Fail	261
Credit Card Validation MO: COR LOC: 56	183	18	3	if (ranges[i].low<= cCardNumber && crCardNumber<= ranges[i].high) <vs> *if(ranges[i].low<= cCardNumber || cCardNumber<= ranges[i].high)*	7 Success 2 Fail	37
Base64 MO: LOR LOC: 147	293	129	2	combined |= b[i + 2] & 0xff <vs> *combined |= b[i + 2] | 0xff*	2 Success 1 Fail	71
Base64 MO: ASR LOC: 144	293	129	2	combined <<= 8 <vs> *combined >>= 8*	2 Success 3 Fail	115

Table 3. Sample of the testing scenarios used for the Credit Card Validation sample program.

```
\mindprod\creditcard\ValidateCreditCard.java 1
F;3799999999999999;vendor_name.equals("Amex")
S;4000000000000;vendor_name.equals("Visa")
S;4999999999998;vendor_name.equals("Visa")
S;0;vendor_name.equals("Err:no enough digits”)
F;6011222233334444;vendor_name.equals("Amex")
```

are available at http://www.cut.ac.cy/staff/andreas.andreou/files and are briefly described below.

Credit Card Validation: This sample program reads a credit card number of x digits and returns its vendor (Visa, AMEX, Diners/Carte Blanche, JCB, MasterCard, or none of the above). The program may also check the digits of the card using an algorithm that is suitable for that particular vendor so as to validate the card number.

Triangle Classification: This program, given the three sides (lengths) of a triangle, performs classification in certain categories (equilateral, isosceles, scalene, not a triangle).

Base 64: This program receives an input string, encodes it using a 64-character set representation and finally decodes it back to string which returns as output.

Table 2 lists the results: the sample program, the mutation performed and the line where the error was located are indicated in the first column, while the subsequent columns include the number of lines in the source code files after normalization, the size in lines of code of the initial slice fed to the algorithm, the number of files that are used to construct the slice, the exact statement(s) with the proposed correction(s) and the number of testing scenarios, both successful and unsuccessful, that were used to guide the evolutionary process. An example of testing scenarios is given in Table 3 for the Credit Card Validation program. Finally, the last column of Table 2 indicates the time needed for the execution of the algorithm.

The error in each case was successfully detected and the correct fix was proposed, with the time frame for providing the solution ranging from 23 s to almost 4.5 min. The time needed to provide a solution depends on the number of lines in the initial slice, the number of the test cases, as well as the number of source files contributing to the formation of the slice. It is logical that the higher the number of both the lines of the slice and the number of files the algorithm needs to visit for creating that slice, the higher the time consumption to reach to locating the fault and propose the correct fix. Also, the more test cases the algorithm has to examine the more time it needs to do so.

5.2 Experiment Series B'

This group includes programs that were fed with single errors corresponding to mutations at the class level. More specifically, the errors induced relate to special features of the Java programming language and also to inheritance. The sample programs used (available at http://www.cut.ac.cy/staff/andreas.andreou/files) were the following:

Person Sorted List: This program receives a set of numbers corresponding to the identity card numbers of persons and sorts them in a list. An object of type Person is inserted in the list if its ID is not already part of it. Finally, it returns the number of persons in the list. The program uses inheritance for different types of persons like for example employee and student. This program was modified in line 12 of the file "Student.java", using the IHD mutation operator, which inserted variable `id` in child class `Student` so as to hide the corresponding variable in the parent class `Person`. Also, this program was used in another experiment where line 83 of the file "Person.java" was modified using operator EOC so as to replace method `equals` with the "==" operand thus providing comparison at the level of reference instead of content.

Shapes: It is a simple program that receives four input parameters (numbers) that correspond to the dimensions of a circle (first parameter – radius), a square (second parameter – side) and a rectangle (third and fourth parameters – sides A and B). The program calculates the area of each shape based on the input parameters and then it sorts the shapes using a Splay Tree according to their area. Finally, it returns the name of the shape with the maximum area. The "Circle.java" file was modified using the IOP operator and more specifically, lines 19 and 20 were interchanged so as to call the method that calculates the area of the shape prior to assigning values to variables `x1` and `y1` which correspond to the sides of the shape.

Table 4. Results obtained using mutations performed at the class level.

Sample Program	Lines of Source Code	Slice Size	Files	Initial Statement VS *Mutated Statement*	Testing Scenarios	Time (sec)
Person Sorted List MO: IHD LOC: 12	430	90	8	`public Student( java.lang.Integer id, java.lang.String n, intag, java.lang.Stringad,java.lang.String p, double g)` <vs> *`public java.lang.Integer id = new java.lang.Integer(0);`* *`public Student( java.lang.Integer id, java.lang.String n, intag, java.lang.String ad, java.lang.String p, double g)`* *<<LINE 12 DELETED>>*	3 Success 2 Fail	20
Shapes MO: IOP LOC: 19, 20	486	85	10	`super.setValues(x1,y1);` `Area();` <vs> *`Area();`* *`super.setValues(x1,y1);`*	4 Success 3 Fail	68
Person Sorted List MO: EOC LOC: 83	430	90	8	`boolean equals = otherId.equals(thisId)` <vs> *`boolean equals = otherId == thisId`*	7 Success 8 Fail	34
Order Set MO: EAM LOC: 260	384	110	2	`int size2 = s2.getSetLast() + 1;` <vs> *`int size2 = s2.getActualSize() + 1;`*	0 Success 2 Fail	35
Graph Shortest Path MO: JTI LOC: 17	859	142	13	`this.cost=cost;` <vs> *`cost=cost;`*	3 Success 2 Fail	107
Graph Shortest Path MO: JSD LOC: 24	859	142	13	`public int scratch=0` <vs> *`public static int scratch=0`*	5 Success 3 Fail	118

Graph-Shortest Path: This program receives a file as input which includes the description of a graph, along with the cost of each edge. More specifically, the program requires three input parameters, the first two corresponding to the start and end of a certain route respectively and the third defining the algorithm which will be used to calculate the shortest path and its corresponding cost. The program returns the sequence of nodes which constitute the shortest path and the cost of that path according to the input values. The program was modified in line 17 of the “Edge.java” file so as to include an error caused by the use of the JTI mutation operator. More specifically, the keyword “`this`” was removed during the assignment of the local variable `cost`. Therefore, the value of `cost` was assigned to the variable itself instead of the local variable `cost`. Also, this program was used in another experiment, where line 24 of the file “Vertex.java” was modified using operator JSD so as to insert the keyword "`static`" in the definition of variable `scratch`, thus causing all instances (objects) of class `Vertex` to have the same value for that variable.

OrderSet: This program receives two sorted arrays as inputs and returns a new array which contains only the elements that are common between the input arrays (line 260 in file "`OrderSet.java`" was modified with the EAMoperator by substituting the call to method `getSetLast()` with the call to method `getActualSize()`).

Table 4 presents the results of the second category of experiments. Again, in all cases the error was successfully detected and corrected. The time of execution ranged from 20 s to something less than 2 min, again depending on the number of lines in the input slice, the number of the test cases and the number of files that contribute to the slice.

Concluding, the two series of experiments provided strong indications that the proposed approach works quite satisfactory, covering a relatively wide variety of errors, both in terms of type and complexity, locating and successfully correcting the erroneous statements in 100 % of the benchmark cases.

5.3 Experiment Series C'

In the last series of experiments we demonstrate the application of the mutation engine on real world source code by selecting source files from he Eclipse IDE following the approach of Nica et al. [26]. We used the same methodology described in that work to load the source code from Eclipse's repository and then we selected some source code files that could demonstrate the applicability of the proposed approach.

The source code files needed to contain classes and methods that could use input parameters from the command prompt and return a type of variable that could be checked automatically (e.g. string or integer). We created a main class for each file which called the module using the parameters from the command line and returned back the result of the class. To investigate how the algorithm scales up we selected a module with less than 80 lines of code, one between 80 and 200 and finally one over 200 lines of code.

Specifically, the Eclipse (2012) modules used were the following:

org.eclipse.core.tools.ByteUtil.java: This module takes as parameters a byte array and an integer and returns back a string with the number of characters being equal to the value of the integer given. In case the string's length is larger than the given integer it appends three dots (…) in the string instead of the rest of the bytes in the array. The error injected to the code was the change of the relational operator in line 15 from `i < bytesToBeShown` to `i <= bytesToBeShown`. The size of this module is 60 lines.

org.Eclipse.core.tools.nsl.integerMap: This module takes as parameters a key and a value so as to store the value in an array of predefined size. If the array is full it proceeds and increases its size. The main method we created for this program receives a number of parameters from the command line and uses them as a pair, with the first value being treated as the key value and the second one as the value to be stored. It then proceeds and calculates the sum of the values stored in the array. The fault injected is located in line 55 where the ++ operator was added in statement

Table 5. Results obtained from Eclipse modules.

Sample Program	Lines of Source Code	Slice Size	Files	Initial Statement VS *Mutated Statement*	Testing Scenarios	Time (sec)
ByteUtil MO: ROR LOC: 15	60	18	1	i<=bytesToBeShown <vs> *i<bytesToBeShown*	2 Success 2 Fail	10
IntegerMap MO: AOIS LOC: 55	90	38	1	values[emptySlot] = ++value;. <vs> values[emptySlot] = value;	2 Success 3 Fail	20
Tool MO: RPR LOC: 126	261	87	1	if (length > 6) vs *if (length <= 6)*	2 Success 4 Fail	57

`values[emptySlot] = value;` resulting in `values[emptySlot] = ++ value;`. The size of this module is 90 lines.
`org.eclipse.jdt.internal.compiler.tool:` This module verifies that the options passed to the compiler tool are valid and stores them in the proper array of string depending on the type of option (zero argument options, one argument options and file manager options) while returning -1 if the option is not valid and 0 or 1 if it is valid. We created a main class which reads as a parameter a given option and checks if it is valid. We then manually injected a fault on line 126 by modifying the statement `if (length <= 6)` to `if (length > 6)`. The size of this module is 261 lines. We used 2 faulty test cases and 4 that verify correctly the program.

Table 5 presents the results of the last category of experiments. Again, in all cases the error was successfully detected and corrected. The time of execution ranged from 10 s to something less than a minute, again depending on the number of lines in the input slice, the number of the test cases and the number of files that contribute to the slice.

The preliminary investigation of the applicability of the proposed approach on "real world" source code from a well-known and widely used IDE may be considered successful and suggests that the engine works nicely for these scenarios as well. Although it concerns only a small sample of programs, the results obtained are quite encouraging, something that indicates that the proposed solution may indeed offer practical benefits to testing real world software code.

6 Discussion on Possible Threads to Validity

This section briefly presents and discusses some considerations:

(i) Some of the programs used in the first two experiment series were small and contained seeded, not real faults, and therefore one may argue that this may affect the validity of the results in some way. This, actually, does not constitute a threat to the validity of the proposed approach as: (a) The programs used were of the order of some hundreds of LOC, which are lower than some studies reported in the literature (e.g. [10, 17]). This size, though, is among the acceptable average sizes for classes and methods within classes. Our approach works at the level of units, therefore the overall size of the program is not so important, as the proposed algorithm will concentrate on the smaller, independent parts of the source code each time. This was exactly the process followed for the larger programs of series C, therefore we consider the proposed approach able to cope with practically any size of code. Additionally, the faults induced simulate the actual omissions or mistakes made by programmers, thus we believe that the impact of the "fakeness" of the errors used in this study is minimal.

(ii) The selection of the testing scenarios was performed manually, based on the type of the program and the relevant functional specifications. The test cases were chosen so as to cover the largest possible number of different paths of execution and this is an issue that deserves research in its own merit; tools that are able to assess the quality of the test cases against the specifications of a program could be used in order to select the best possible set of test data automatically; this is something we plan to pursue in the near future.

(iii) One of our concerns was the fact that a change being made to a part of the code could influence other areas of the program possibly creating new (regression) faults in the case where multiple errors exist in the code. Here we may identify two different scenarios: (a) The fault repaired affects a line of code that is below the changed statements as we move from top to bottom in the sequence of execution; this would not really be considered a problem as the new fault would be identified at a subsequent stage in a new slice, (b) The algorithm changes a previously repaired statement that removed a formerly addressed fault so this alteration goes undetected. At the moment, in order to handle this problem, we create a log-file that contains all previous changes (repairs) made in the code; a controller module consults this log-file prior to exercising mutations and prevents the algorithm from attempting to make any changes to repaired statements. In this case the algorithm skips the error, puts it in a separate file with "uncorrected" faults and continues with the next fault with proper notification of the user.

(iv) A final possible threat is that the slicing tool works with Java ver. 5 but not with newer versions, which means that we had to use the corresponding JRE in order to create the slice for the program under testing. No tools make use of newer Java versions at the moment, but we plan to tackle this issue by either modifying existing tools to work with newer Java versions, or actually implementing our own tools.

7 Conclusions

Software development suffers from low product quality and high percentage of project failure. The presence of faults is one of the major factors affecting the quality of delivered products; that is why a high percentage of development time is devoted to testing. Most of the time spent on software testing is devoted to actually locating the faults in the source code instead of correcting them. It is obvious that there is a strong need to develop a highly effective method for fault detection so as to reduce the time required by the testing process and assist in increasing the quality of the code and the productivity of software developers.

In this work we proposed a novel approach that is able to automatically detect and correct faults in Java code. The approach utilizes backward static code slicing for localising a fault and suggests possible corrections with the use of Mutation Testing. Indus/Kaveri was used for creating the slices, while MuJava was the tool adopted for applying different mutation operators selected by the user, both at the method, as well as at the class level. The process of fault detection and correction through statement replacement is a difficult problem to tackle with a large solution space; thus, we resorted to using Genetic Algorithms so as to reduce it to a search optimization problem. The GA evolved a number of candidate solutions-replacements of statements that were assessed by a dedicated fitness function.

Two series of experiments with hand-seeded errors were conducted using sample programs corresponding to well-known problems that are normally used as benchmarks for testing. Series A' involved programs with method-level errors, specifically arithmetical, relational, conditional, logical and assignment errors, while series B' included programs containing class-level errors related to Java features and inheritance. The results suggested that the proposed approach works quite satisfactory, covering a wide range of errors, both in terms of type and complexity. In addition, the engine always identifies the error and suggests the correct replacement that removes the fault.

A third series of experiments was also performed on real-world software, using modules of the Eclipse IDE. The aim here was to assess the efficacy of the proposed approach using source code of a very familiar program. We manually inserted faults in different parts of the code of three modules selected and the engine yielded the correct replacements to remove them.

The validation results of the experimentation process suggested that the proposed approach is quite successful leading to the provision of applicable tools that may greatly enhance the effectiveness of the testing process.

There are quite a few research steps that may be performed based on the present work:

(i) We will attempt to collect more real case examples of programs from open source repositories so as to have a richer close-to-reality set of results.
(ii) We plan to investigate the potentials of integrating a supporting module to the existing tool that will enable a more "sophisticated" way for selecting the appropriate test case scenarios automatically based on pre- and post-conditions that express the functional specifications of a program. This would be made possible with the use of the Java Modeling Language (JML) that is used to describe such specifications in code. It uses "requires to specify" post-conditions, that is, what needs to be true to call a method, and "ensures to specify" post-conditions, that is, what needs to be true when the function terminates. They are

included in the source code file just before the declaration of methods and are being compiled along with the normal source code which essentially will be checked against. JML may also aid in limiting the number of proposed solutions if more than one exist, again based on the pre-conditions and post conditions found in the source code.

(iii) We will address the problem of regression faults by attempting to provide simultaneous repairs to more than one fault. To this end we plan to modify our Genetic Algorithm so as to support multi-objective optimization through multithreading and parallel processing.

(iv) The tool will be upgraded to include statements of Java that our current implementation of the parser does not support. This will enable us to cover an even larger number of errors by increasing the set of mutation operators supported and including more complicated replacement operators. Also, direct calls to the Indus tool will be provided, without the need for support by the Eclipse platform.

(v) Last, we will focus on upgrading existing or developing new tools that support Java code of newer versions while supporting newer features, like JUnit test cases, to assist in the assessment of possible solutions. Also, newer mutation tools, better optimized for today's processors, will be integrated in the proposed engine to provide solutions to some other mutation problems like equivalent mutants.

References

1. Patton, R.: Software Testing, 2nd edn. Sams Publishing, Indianapolis (2006)
2. Myers, G.J.: The Art of Software Testing. Wiley-Interscience, New York (1979)
3. Weiser, M.D.: Program slices: formal psychological and practical investigations of an automatic abstraction method. Ph.D. thesis, University of Michigan, USA (1979)
4. Reps, T., Bricker, T.: Illustrating interference in interfering versions of programs. In: Proceedings of the Second International Workshop on Software Configuration Management (1989)
5. Agrawal, H., Horgan, J.R., London, S., Wong, W.E.: Fault localization using execution slices and dataflow tests. In: Sixth International Symposium on Software Reliability Engineering, Vol.24(27), pp. 143–151 (1995)
6. Agrawal, H., Horgan, J.R.: Dynamic program slicing. In: ACM SIGPLAN Conference on programming Language Design and Implementation, White Plains, New York, U.S.A., pp. 246–256, ACM Press (1990)
7. Black, S., Counsell, S., Hall, T., Wernick, P.: Using program slicing to identify faults in software. In: Dagstuhl, Seminar N°, vol. 05451 (2005)
8. Hamlet, R.G.: Testing programs with the aid of a compiler. IEEE Trans. Softw. Eng. **3**(4), 279–290 (1977)
9. DeMillo, R.A., Lipton, R.J., Sayward, F.G.: Hints on test data selection: help for the practicing programmer. IEEE Comput. **11**(4), 34–41 (1978)
10. Harman, M., Jia, Y., Langdon, W.B.: Strong higher order mutation-base test data generation. In: ESEC/FSE'11, Szeged, Hungary, 5–9 Sept 2011
11. Nica, M., Nica, S., Wotawa, F.: On the use of mutations and testing for debugging. Softw.-Pract. Exp. (2012) (Article published online)

12. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Upper Saddle River (1989)
13. Jiang, T., Gold, N., Harman, M., Li, Z.: Locating dependence structures using search based slicing. J. Inf. Softw. Technol. **50**(12), 1189–1209 (2008)
14. Arcuri, A., Yao, X.: A novel co-evolutionary approach to automatic software bug fixing. In: IEEE Congress on Evolutionary Computation IEEE World Congress on Computational Intelligence, pp. 162–168 (2008)
15. Weimer, W., Forrest, S., Le Goues, C., Nguyen, T.: Automatic program repair with evolutionary computation. Commun. ACM Res. Highlight **53**(5), 109–116 (2010)
16. Offutt, A.J.: The coupling effect: fact or fiction. In: ACM SIGSOFT ‘89—Third symposium on Software testing, analysis, and verification. ACM, New York (1989)
17. Fraser, G., Zeller, A.: Generating parameterized unit tests. In: International Symposium on Software Testing and Analysis (ISSTA’11), Toronto, Canada, 17–21 July 2011
18. Offutt, A.J., Untch, R.H.: Mutation 2000: uniting the orthogonal. In: 1st Workshop on Mutation Analysis (MUTATION’00), San Jose, California, pp. 34–44 (2001)
19. King, K.N., Offutt, J.: A Fortran language system for mutation-based software testing. Softw. Pract. Exp. **21**(7), 686–718 (1991)
20. Binkley, D., Danicic, S., Gyimothy, T., Harman, M., Kiss, A., Korel, B. Theoretical foundations of dynamic program slicing. Theoretical Computer Science, **360**, 23–41 (2006)
21. Offutt, J., Ma, Y.S., Kwon, Y.R.: MuJava: an automated class mutation system. Softw. Test. Verif. Reliab. **15**, 97–133 (2005)
22. Ma, Y.S., Offutt, J.: Description of method-level mutation operators for Java. (2005) http://cs.gmu.edu/~offutt/mujava/mutopsMethod.pdf
23. Wang, T., Roychoudhury, A.: Using compressed bytecode traces for slicing Java programs. In: ACM/IEEE International Conference on Software Engineering (ICSE), pp. 512–521 (2004)
24. DeYong, J.: Program slicing for Java 1.6. Master Thesis, Eindhoven University of Technology, Eindhoven, Holland (2011)
25. Yiasemis, P., Andreou, A.: Dynamic Localisation and Automatic Correction of Software Faults Using Evolutionary Mutation Testing. In: Cordeiro, J., Maciaszek, L.A., Filipe, J. (eds.) ICEIS 2012. LNBIP, vol. 141, pp. XX–YY. Springer, Heidelberg (2012)
26. Nica, S., Ramler, R., Wotawa, F.: Is mutation testing scalable for real-world software projects? In: VALID Third International Conference on Advances in System Testing and Validation Lifecycle, Barcelona, Spain (2011)

Towards a Leaner Evaluation Process: Application to Error Correction Systems

Arnaud Renard[1,2(✉)], Sylvie Calabretto[1,2], and Batrice Rumpler[1,2]

[1] Universit de Lyon, CNRS, Lyon, France
{arnaud.renard, sylvie.calabretto, beatrice.rumpler}@liris.cnrs.fr
[2] INSA-Lyon, LIRIS, UMR5205, 69621 Villeurbanne Cedex, France

Abstract. While they follow similar procedures, evaluations of state of the art error correction systems always rely on different resources (collections of documents, evaluation metrics, dictionaries, ...). In this context, error correction approaches cannot be directly compared without being re-implemented from scratch every time they have to be compared with a new one. In other domains such as Information Retrieval this problem is solved through Cranfield like experiments such as TREC [5] evaluation campaign. We propose a generic solution to overcome those evaluation difficulties through a modular evaluation platform which formalizes similarities between evaluation procedures and provides standard sets of instantiated resources for particular domains. While this was our main problem at first, in this article, the set of resources is dedicated to the evaluation of error correction systems. The idea is to provide the leanest way to evaluate error correction systems by implementing only the core algorithm and relying on the platform for everything else.

Keywords: Evaluation model · Framework · Error correction · Textual documents · Distance and similarity measure · Metrics · Information retrieval

1 Introduction

In order to propose an accurate way to evaluate error correction systems, it is interesting to pay a special attention to their benefits and particularly to the different shape of errors they have to deal with. Indeed, an error correction system will be susceptible to perform worse or better according to the type of errors the system will have to face to. So, this will have to lead to different evaluation mechanisms. In this paper, we are particularly concerned by error correction of a specific subset of data which consists in textual data. Indeed, large amounts of data produced every day by the growing number of the Web 2.0 services users are error-prone. It makes it important to correct those errors while they may disturb data management applications. While spell checkers are amongst the most common Natural Language Processing (NLP) applications, many computer applications rely on clean text processing techniques. It is only

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 228–242, 2013.
DOI: 10.1007/978-3-642-40654-6_14,

because of the increase of noisy text [16] that these techniques have been adapted to take noise like errors into account.

Most documents were formerly produced by professionals who have to keep a minimum level of quality while writing. Indeed their writings have to conform to quality controls like newspaper editorial chain, article review... At Web scale, the way information is produced is different while most (but not all) documents are created by ordinary users [13]. In this last case, information is not provisioned as a result of a professional work. Ordinary users are more likely to make mistakes while using an inappropriate terminology (or a vocabulary they are not familiar with). It is therefore legitimate to have some reservations about the quality of their writings (both about the form and the substance). Moreover, web published content is not constrained by quality control. For example, weblogs have popularized the mass self-publishing with free and immediate release.

According to the problem of information quality, it might be interesting to consider errors in Information Retrieval systems (IR) as well while it is one of the principal ways to access data on the Web. Most of the time attempts to correct errors with an IR improvement perspective consider only query correction like the popular "Did you mean". There are few researches aiming to correct documents themselves like Ruch works [14], and works related to former TREC-5 Confusion Track [5] with OCR related errors and later TREC-6 with Spoken document retrieval [18] track. However, at web scale, it is an important area of improvement for IR systems [17]. Our state of the art led us to identify difficulties in the benchmarking of error correction systems. For example there is no common (and realistic) evaluation collection and some [1,12] publish their testing sets while many others do not. It is important to have common testing environments which rely on common collections and standardized metrics in order to be able to compare solutions.

Our proposal consists in an evaluation model which applies to error correction systems as well as low-level resources they rely on. For our sake, evaluations results obtained later thanks to the framework implementing this model through our platform will then allow the choice of the "best" error correction system to use in the indexation phase of an IR system. That is why the specialization of our model exposed later will be particularly focused on this aspect.

The context and positioning of this article is presented in Sect. 2 which defines the key concepts used along the article and establishes a classification of common errors. In the Sect. 3 we present an overview of different error correction approaches as well as practical issues related to the difficulties of evaluating them. In order to address this problem, Sect. 4 presents our evaluation model across a generic meta-model which is derived in a model we use to evaluate error correction systems. The evaluation platform implementing this model is presented in Sect. 5. It allowed the analysis of some error correction mechanisms. Implemented resources are described in Sect. 6 as well as our first evaluation results. Finally, Sect. 7 provides our conclusions on the evaluation of error correction systems and presents our perspectives for their future integration into IR systems.

2 Context and Positioning

According to Shannon works related to information theory [15], noise can be described as a corruption of information resulting in a difference between the expected information (which is supposed to be correct) and the information obtained (which might contain errors). At first, it is important to define what an error is, and at least to clarify the definition retained in this article.

2.1 Key Concepts Definitions

Alphabet. If we consider textual information and take A, a finite set which we call *alphabet* (in the case of the English language, A is matching all possible characters in English). Thus, every *character* c belongs to the alphabet A, ($c \in A$).

Word. Let A^k be the set of *words* w composed by a sequence of k ordered characters.

$$w \in A^k \Leftrightarrow w = c_1, c_2, \ldots, c_{k-1}, c_k \ . \tag{1}$$

Dictionary (or Lexicon). We call *dictionary* d (or *lexicon*), all valid words of a language coming from an alphabet A (i.e. currently or formerly used by native speakers of the language).

Error. An *error* e can be defined as the presence of at least one character which differs from the expected character at a given position in the sequence corresponding to a word w. Let w be in A^k and $c_i(w)$ denote the character at position i in w :

$$w_1, w_2 \in A^k : w_1 \neq w_2 \Leftrightarrow \exists i : c_i(w_1) \neq c_i(w_2) \ . \tag{2}$$

This definition covers all errors like ones due to the insertion, the deletion, or the substitution of a character by another one as well as any other operation which modify the sequence of characters composing a word.

Wrong Word (Resp. Target Word). We define a word with at least one error (as w_2) as a *wrong word* different from the correct intended *target word* (as w_1).

Error Correction System. ording to previous definitions, an *error correction system* is a mechanism which allows to retrieve the correct intended target word corresponding to a wrong word.

This preliminary definition of an error stays at high level so that it is possible to refine it like we propose in the following section.

2.2 Taxonomy of Errors

Errors in digital documents may have multiple origins. Indeed, errors can occur (and accumulate) at each step in the process which leads to an electronic document from an initial "abstract" information. Information is qualified as abstract as it is not materialized in a physical medium digital or not. It is then possible to distinguish digital documents whether they are produced from a direct materialization (e.g. keyboard input), or they passed through another state (e.g. handwritten) before being digitized. We can make a distinction between errors related to human intervention during information creation (false initial idea), expression (verbalization of idea, association of idea with a word, spelling or pronunciation problem), and writing of information (dysgraphia, poor typography), and secondly, errors coming from computer data processing which occurs during OCR phase.

In some cases, errors can be *valid words* as defined above. This type of error is called *real-word error* (e.g. "diary" and "dairy"). Although this type of error preserves most of the time the "syntactic" validity of the sentence in which it occurs, it breaks its semantic coherence making the sentence unintelligible by humans. Such errors cannot be detected (and hence corrected) efficiently without the presence of a context such as words adjacent to the error. The context makes it possible to identify semantic inconsistencies generated by the error, or at least the low statistical probability for this word to be surrounded by the words which compose its context. In most cases, errors result in *invalid words* that we call *non-words errors* (e.g. "tree" and "teer"). This last type of error is easier to detect because a simple comparison with valid words of a dictionary is sufficient. Although the presence of a context may help to identify more precisely a proper correction it is not mandatory while these errors can be considered as isolated words out of context.

Most real-words errors come from abstract information, i.e. there is difficulty in the ability of author to associate the correct word according to his thought (see Fig. 1). This problem occurs most frequently with children, non-native writer, and dyslexic people. However, typos and errors in character recognition rarely belong to real-words.

The different types of errors being defined, we offer an overview of approaches to correct them according to their type.

3 Considerations About Evaluation of Error Correction Systems

In this section, we will introduce the most important approaches developed for the correction of errors. While this is a very active topic, it is not an exhaustive state of the art in terms of references. Nevertheless, we believe that great families of approaches are represented.

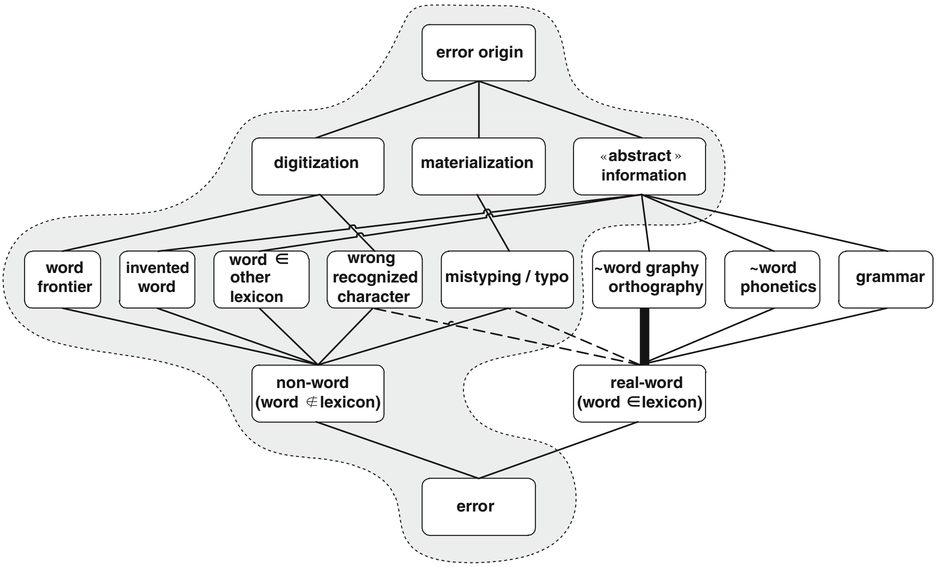


Fig. 1. Multi facet errors classification (error origin/type).

3.1 Overview of Error Correction Approaches

In this article, we do not choose to consider one type of error (non-word or real-word) in particular even if the latter one is more difficult to identify as an error.

Works on non-words error correction are referenced by [6,9]. However, approaches having the best results rely on context, as well as approaches to correct real-words errors.

Works on real-words error correction can be classified into two categories: methods based on semantic information (or a human lexical resource) and methods based on machine learning (or information likelihood).

Approach based on "information semantics" was first proposed by [4], and developed later by [3]. It detects semantic anomalies but is not limited to word verification from predefined confusion sets (at least pairs of commonly confused words) which model ambiguity between words. This approach is based on the observation the words the writer intends to write are usually semantically related to surrounding words while some errors resulting real-words are not. The problem of detecting real-word errors is the same as the problem of homonyms. This is an application of disambiguation methods to correct errors.

Mays et al. [7] propose a statistical method using probabilities of trigrams of words to detect and correct real-words errors without requiring predefined confusion sets. Wilcox-O'Hearn et al. [21] analyze advantages and limitations of the method proposed by [7]. They present a new evaluation of the algorithm in order to be able to compare the results with other methods. They also built and evaluated some variants of the algorithm using fixed-size windows.

3.2 Problems with Evaluation of Error Correction Systems

All these works refer to difficulties in the evaluation of their approaches compared to the others. That is why works such as those of [21] are very important. Used resources (reference dictionary, collections of errors, evaluation metrics) differ significantly from the evaluation of one approach to another. Thus, collections of errors (or collections of documents which contain errors) are rarely employed in the evaluation and most of the time based on randomly generated errors in a collection of documents. A significant work from Pedler [12] has been to collect and make available documents produced by dyslexic people.

We propose a flexible evaluation model adapted for our needs to the evaluation of error correction mechanisms. However, it could relatively easily be adapted to evaluate other kind of systems.

4 Proposal of an Evaluation Model

In order to allow a maximum level of re-usability, we have defined a generic approach to evaluate systems. It may be closed systems considered as black boxes as well as composites systems created from an original combination of resources to evaluate.

This evaluation approach is described at a macroscopic level by a meta-model we call the *Generic Evaluation Model* (GEM). Our main concern in the context of this paper is the evaluation of different error correction mechanisms. So, we rely on a *Specific Evaluation Model* (SEM) derived from the GEM and adapted to this case. The SEM is tuned to evaluate the wanted type of system and only needs to be instantiated to perform an experiment.

4.1 Definition of a Generic Evaluation Model (Meta-Model)

The GEM is a generic abstract representation of an evaluation model which consists of five elements, so that the GEM can be defined by the 5-tuple:

$$GEM = \langle R_D, R_P, s, R_E, a \rangle \ . \tag{3}$$

Where R_D, R_P and R_E are input resources families to the model. These resources respectively belongs to the following families:

- Data D: noted R_D (e.g. data to process),
- Processing P: noted R_P (e.g. algorithms to apply to data),
- Evaluations E: noted R_E (e.g. evaluation metrics, reference values).

Each resource family includes a set of types of resources of its own and is dependent on the derivation of the GEM in SEM.

s is a data processing module based on the resources R provided to produce results (e.g. scores).

e is a module to evaluate data processing s results and produces performance indicators (e.g. accuracy).

This meta-model is too generic to be usable for evaluation task. It must be instantiated in a specific model SEM defined relatively to an experiment evaluation needs.

4.2 Derivation of a Specific Evaluation Model for Evaluation of Error Correction Systems (Model)

The SEM is a derivation of the GEM for the needs of a particular evaluation. In this paper, it has been derived to evaluate error correction mechanisms. These can be full autonomous error correction systems which have their own resources (this is a special case which will be specified later), or composite systems as mentioned above. To define the SEM, we will initially define each family of resources based on resource types it accepts.

Thus R_D consists of resources r_t of type $Coll$ and $Dict$. Where, $Coll$ represents the type Collection of documents which is represented by a list of pairs of the form: $wrongword, targetword$. And $Dict$ represents the Dictionary type which is a list of the form: $word, wordfrequency$.

Similarly, R_P consists of resources r_t of type SDM or AS. The use of one of these two types excludes the use of the other type of resource. Where, SDM represents the type Similarity and Distance Measure whose values are normalized in $[0, 1]$ interval. While employed measures are standardized, the similarity is $1 - distance$ and vice versa. And AS, is a Autonomous error correction System.

Finally, we can define R_E as resources r_t of type EM. Where EM represents the type Evaluation Metrics whose values are normalized in $[0, 1]$ interval.

Each family of resources is subject to constraints on its cardinality which can be different if the evaluated error correction system is autonomous or composite.

Thus, evaluation of a composite system (the general case) requires the instantiation of a resource of each type:

$$\forall t \in \{Coll, Dict, SDM, EM\}, |r_t| \geq 1 \ . \tag{4}$$

The SEM is then represented by the following 5-tuple:

$$SEM_{composite} = \langle \{Coll, Dict\}, SDM, s, EM, a \rangle \ . \tag{5}$$

However, when evaluating an autonomous system, it is considered as a processing resource AS instead of SDM. In addition, the system is autonomous, and does not require any dictionary.

The SEM is then represented by the following 5-tuple:

$$SEM_{autonomous} = \langle \{Coll\}, AS, s, EM, a \rangle \ . \tag{6}$$

The proposed model formalizes concepts and follows intuitive evaluation logic. However, this formalization is necessary for large scale evaluation. The genericity of the model enables it to apply to the evaluation of various types of systems via the instantiation of suitable resources. In this case, the model was adapted to evaluate error correction mechanisms. The model was then implemented in a platform which can serve as a framework for evaluation.

5 Implementation of the Evaluation Model

The implemented evaluation platform is based on the above model which defines its different modules.

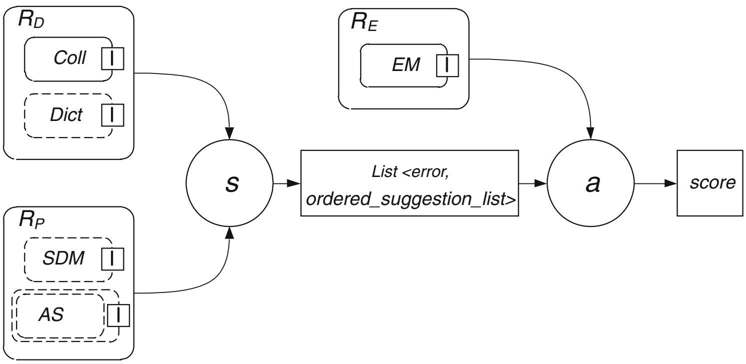


Fig. 2. Evaluation model and overall architecture of the evaluation platform.

The platform was developed in Java and uses the OSGi standard [11] for modules implementation. This allowed us to use the modularity of the proposed model by defining common standard interfaces for each type of resources. This makes it possible for a given type of module to replace it easily without impacting the rest of the platform. Each module respect a contract has its own life cycle and can be dynamically deployed on the platform. Processing module s and assessment module a ensure the availability of the minimum needed set of resources for testing.

The developed platform was used for our evaluation of some composite systems built from dictionaries, similarity (or distance between strings) measures commonly used in error correction systems (Fig. 2).

6 Evaluation

6.1 Instantiation of Evaluation Model Resources

Evaluations conducted in this article consider only a reduced set of composite systems. Resources used in the composition of these systems are exposed in following paragraphs.

Errors Collection. Our evaluations only concern errors corrected regardless of any context (at first time). This collection of errors has been compiled from common mistakes on Wikipedia, Wikipedia List of Common Misspellings WCM [19]. Errors come as a list of 4408 couples of the form:

```
<wrong word,target word>
```

This collection contains non-words as well as real-words errors. While real-words errors are already labelled as errors, it is possible to suggest a correction without the need for a context (which is not available).

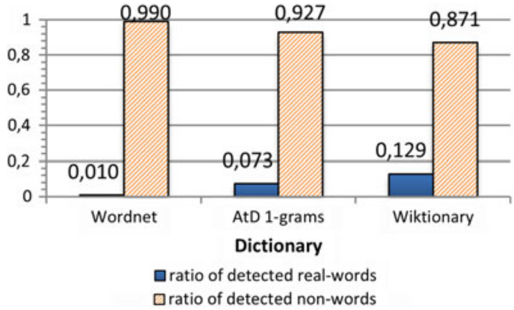


Fig. 3. Proportion of words in the collection identified as real-words (resp. non-words) according to the used dictionary.

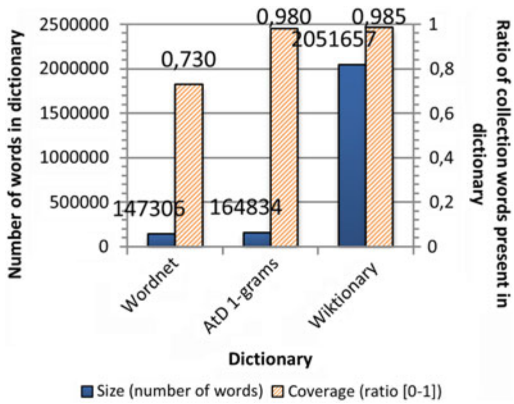


Fig. 4. Dictionaries size and collection of errors target words coverage.

Dictionary. In our first experiments, we implemented three different dictionaries. A dictionary based on Wordnet [2,8], a unigram dictionary provided by the AtD system [10], and an online collaborative dictionary Wiktionary [20].

Although a correction can be proposed both for non-words and real-words errors, identification of the belonging of these errors to one or other of the categories is interesting to segment the collection and provide independent indicators. This is a difficult task because the identification of the category is dependent of the chosen dictionary (Fig. 3). Thus, new words not yet integrated in a dictionary can be wrongly considered as non-words while unusual words may persist. The temporal aspect is difficult to manage.

It may be noted on the histogram above that a larger dictionary tends to identify more errors as real-words errors than a dictionary with fewer words. In that sense, Figure 4 highlights the difficulty to choose a dictionary.

Indeed, the Wordnet based dictionary contains 147,000 words, and covers only 73 % of target words corresponding to errors, while AtD dictionary has

Table 1. Synthesis of online and offline error correction systems constraints.

	Online error correction (standard)	Offline error correction (ITEC)
Contextual data	Yes: directly usable	No: metadata assumptions
Users interactions	Yes: choice among many proposals (5)	No: no choice high accuracy required

coverage of 98 % with nearly 165,000 words only. Wiktionary dictionary has coverage of about 98.5 % with over 2 million words.

AtD dictionary leverage the coverage of target words in the collection of errors according to the number of words in the dictionary. Indeed it is the result of a learning phase which allowed to keep only most frequent words.

Similarity and Distance Measures. As part of a first series of experiments, only three similarity/distance measures were evaluated: Levenshtein distance, Jaro and Jaro-Winkler distance. These three measures will then have to be compared and maybe combined with other similarity measures such as phonetic encoding based measures.

Evaluation Metrics. As a perspective we would like to integrate an error correction system to an information retrieval system in order to improve its performance *Indexation Time Error Correction* (ITEC) process described in further works of Sect. 7). If the error correction process is conventional, it is necessary to differentiate interactive or *online* error correction, and non-interactive or *offline* error correction.

Indeed, in the case of online error correction, the system benefits from contextual information about user input according to the device (smartphone, tablet, netbook, laptop) and the physical layout of the keyboard keys. Moreover, it is possible to suggest multiple corrections to the user at the same time. This allows the user to choose the suggestion which fits best. Therefore, it is more important to suggest the appropriate correction among propositions rather than ranking it at the first position among them.

In the case of offline error correction, the problem is more complex. Indeed, no contextual information about input of data (and available metadata are relatively poor). Nevertheless, it is possible to assume that English texts should be linked to standard QWERTY keyboard layout. It is particularly important to promote accuracy in the case of offline error correction systems. Indeed, it is important to suggest the proper correction in first place because the system cannot rely on a user to choose the final correction. This phenomenon is necessarily present when one wishes to reduce the responsibility of the user. It is therefore necessary to have a correction system with maximum accuracy.

These characteristics guided our evaluation metric choice to the Mean Reciprocal Rank noted MRR [18]:

$$MRR = \frac{1}{|errorCouples|} \sum_{i=1}^{|errorCouples|} \frac{1}{sugTWR_i} . \tag{7}$$

Where $sugTWR$ stands for the rank of the suggestion which is effectively the same as the target word.

This metric seems to be suited to the constraints of offline error correction evaluation. Indeed, the MRR applies a significant penalty if the correct result does not occur in first ranks. High MRR value means that the correct result belongs to top ranked results. On the opposite, a low MRR value doesn't mean that the correct result is ranked very far, but only not in the first ones.

Instances of employed resources being defined, the next section presents the results of initial experiments.

6.2 Experimental Results

Instances of previously defined resources allowed us to build a composite error correction system to be evaluated for each combination dictionary/similarity measure, nine systems found in nine Evaluation Model Instances rated EMI:

$$\begin{aligned}
EMI_1 &= \langle\{WCM, Wiktionary\}, Jaro - Winkler, s, MRR, a\rangle \\
EMI_2 &= \langle\{WCM, Wiktionary\}, Jaro, s, MRR, a\rangle \\
EMI_3 &= \langle\{WCM, Wiktionary\}, Levenshtein, s, MRR, a\rangle \\
EMI_4 &= \langle\{WCM, AtD\}, Jaro - Winkler, s, MRR, a\rangle \\
EMI_5 &= \langle\{WCM, AtD\}, Jaro, s, MRR, a\rangle \\
EMI_6 &= \langle\{WCM, AtD\}, Levenshtein, s, MRR, a\rangle \\
EMI_7 &= \langle\{WCM, Wordnet\}, Jaro - Winkler, s, MRR, a\rangle \\
EMI_8 &= \langle\{WCM, Wordnet\}, Jaro, s, MRR, a\rangle \\
EMI_9 &= \langle\{WCM, Wordnet\}, Levenshtein, s, MRR, a\rangle
\end{aligned} \tag{8}$$

Figure 5 shows MRR scores obtained by each of instantiations of the model. As it can be seen, bigger dictionary (see Fig. 4) as Wiktionary allows a maximum coverage of target words in the collection of errors at the cost of a lower rank of the target word among the suggestions.

Indeed, the correct target word is lost in the quantity of words syntactically close to the misspelled word, which leads to a low MRR. A dictionary of smaller size allows a better ranking of the target word at the cost of an increased risk that suggestion list misses the target word.

If we consider string similarity measures, we can see they have different behaviors according to the dictionary (although they seem to be close most of the time). Thus, Levenshtein seems to be the least sensitive of the three to the size of the dictionary, while Jaro-Winkler which obtained good results associated with Wordnet (small dictionary) seems to be less effective when combined

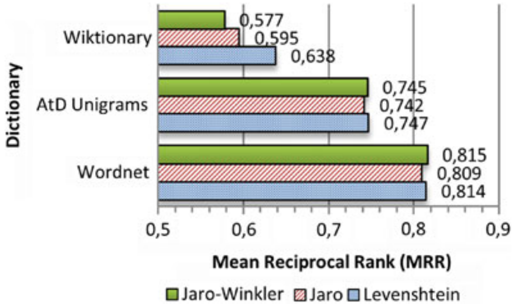


Fig. 5. MRR of different combinations between similarity measures and dictionaries.

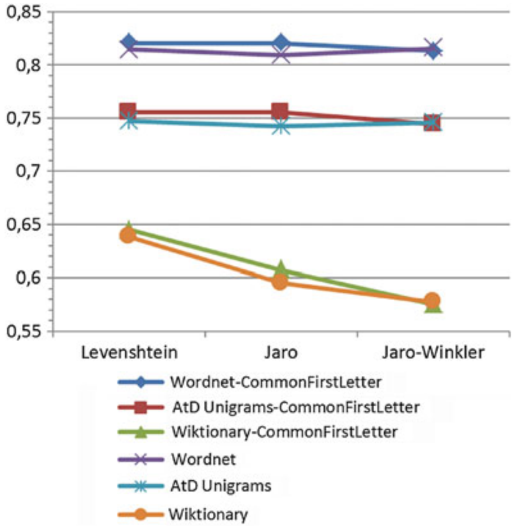


Fig. 6. Comparison of MRR values obtained with and without taking into account a common first character.

with Wiktionary. The difference between these measures is not very important because they are not fundamentally different.

The study of the WCM collection allowed us to determine that among the 4408 couples that the collection contains 4274 wrong words share their first character with their associated target word. It means that 97 % of errors couples share their first character. We modified previously used similarity measures so that they return a null similarity to dictionary words which do not share the same first character as the misspelled word to correct. The results of nine new EMI are shown in Fig. 6.

We can observe from the Fig. 6 that the heuristic which consists in the use of the first character slightly improves the MRR of two of the three measures. Only the Jaro-Winkler measure sees its MRR lowered. Moreover, this heuristic reduced significantly the computing time of suggested corrections by eliminating a large number of candidate words each time an error is processed. This heuristic thus seems interesting to integrate in composite systems.

In order to evaluate different kind of similarity measures, we decided to apply previous string similarity measures on phonetic encodings of both errors and candidates words. This allows the creation of a phonetic similarity measure. We can observe on Fig. 7 that the combination of both measures is getting worse results than simple string similarity measures (about half the MRR of sring similarity measures).

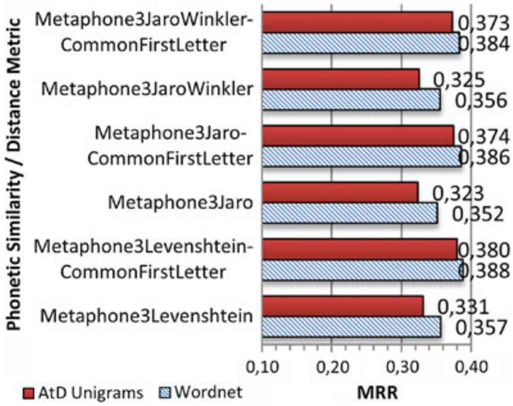


Fig. 7. Comparison of MRR values obtained with a combination of String similarity measures applied over a Metaphone 3 phonetic encoding.

This can be explained by the fact that the phonetic encoding made many word candidates to be encoded by the same phonetic key. The problem comes from the pessimistic computation of the rank of the correct result. Indeed, in the case where many candidate words obtain the same score after the scorer pass, our assessor consider that the rank of the correct result is the rank of the worst one. So, if the ten best words candidates including the correct result have the maximum score of 1, the assessor will consider that its rank is 10, not 1, not 5. It should be better to be more fair in this case by using word frequency as a second criterion to sort the results (or in the worst case by putting the rank of the correct word at the mean rank of the same scored candidates).

7 Conclusions and Further Works

In this paper, we proposed a formal definition of key concepts related to error correction. We also proposed a classification of these errors according to their origins and their types and their related difficulties. Our state of the art about error correction systems allowed us to identify a problem in the evaluation of these systems. We have proposed a comprehensive evaluation model including a meta-model derived in a model that we instantiated. Afterwards, this evaluation model was implemented in a modular and extensible evaluation platform we used to evaluate 18 instances of the model through composite systems. While this is not sufficient to validate the model in itself, it is hard to provide a meta-evaluation with regards to other evaluation approaches. It only proves that it works for evaluated cases.

As the developed platform is extensible we will integrate other similarity measures between strings, as well as phonetic similarity measures. We will also incorporate other heuristics such as those proposed in [22]. Other collections of errors such as the one used by Aspell [4] will be included as well as collections of documents tagged with errors such as the one used by [12]. The platform can then be used to determine optimal parameters in the combination of different approaches and heuristics. We wish to evaluate complete error correction systems on the same platform. The results are more difficult to interpret because we do not control the resources (including dictionaries) they rely on, but they will provide reference results to locate raw performance of the evaluated approaches.

An Indexation Time Error Correction (ITEC) system can be used in the analysis of documents to correct errors they contain and allowing creation of more representative indexes. We wish to make indirect evaluation of error correction approaches by comparing the results obtained by information retrieval systems on evaluation campaigns such as TREC [5] or INEX without ITEC and with it enabled.

References

1. Atkinson, K.: Aspell Spellchecker. http://aspell.net (2012). Accessed 15 Jan 2012
2. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
3. Hirst, G., Budanitsky, A.: Correcting real-word spelling errors by restoring lexical cohesion. Nat. Lang. Eng. **11**(1), 87–111 (2005)
4. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms, Chapter 13. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, vol. 305, pp. 305–332. MIT Press, Cambridge (1998)
5. Kantor, P.B., Voorhees, E.M.: The TREC-5 confusion track: comparing retrieval methods for scanned text. Inf. Retrieval **2**(2), 165–176 (2000)
6. Kukich, K.: Techniques for automatically correcting words in text. ACM Comput. Surv. (CSUR) **24**(4), 439 (1992)
7. Mays, E., Damerau, F.J., Mercer, R.L.: Context based spelling correction. Inf. Process. Manag. **27**(5), 517–522 (1991)
8. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
9. Mitton, R.: Ordering the suggestions of a spellchecker without using context. Nat. Lang. Eng. **15**(02), 173–192 (2008)
10. Mudge, R.: After the Deadline. http://static.afterthedeadline.com (2012). Accessed 15 Jan 2012

11. OSGi-Alliance. Open Services Gateway initiative. http://www.osgi.org (2012). Accessed 15 Jan 2012
12. Pedler, J.: Computer correction of real-word spelling errors in dyslexic text. Ph.D. thesis, Birkbeck, London University (2007)
13. Rosnay, J., Revelli, C.: Pronetarian Revolution (2006)
14. Ruch, P.: Using contextual spelling correction to improve retrieval effectiveness in degraded text collections. In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, p. 7. Association for Computational Linguistics (2002)
15. Shannon, C.: A mathematical theory of communication. Bell Sys. Tech. J. **27**(379–423), pp. 623–656 (1948)
16. Subramaniam, L.V., Roy, S., Faruquie, T.A., Negi, S.: A Survey of Types of Text Noise and Techniques to Handle Noisy Text. Language, pp. 115–122 (2009)
17. Varnhagen, C.K., McFall, G.P., Figueredo, L., Takach, B.S., Daniels, J., Cuthbertson, H.: Spelling and the web. J. App.l. Develop. Psychol. **30**(4), 454–462 (2009)
18. Voorhees, E.M., Garofolo, J.: The TREC-6 spoken document retrieval track. Bull. Am. Soc. Inf. Sci. Technol. **26**(5), 18–19 (2000)
19. Wikipedia Community. Wikipedia List of Common Misspellings. http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings (2012). Accessed 15 Jan 2012
20. Wiktionary Community. Wiktionary Online Collaborative Dictionary. http://en.wiktionary.org/wiki/Wiktionary:Main_Page (2012). Accessed 15 Jan 2012
21. Wilcox-O'Hearn, A., Hirst, G., Budanitsky, A.: Real-word spelling correction with trigrams: a reconsideration of the Mays, Damerau, and Mercer model. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 605–616. Springer, Heidelberg (2008)
22. Wong, W., Liu, W., Bennamoun, M.: Integrated scoring for spelling error correction, abbreviation expansion and case restoration in dirty text. In: 5th Australasian conference on Data mining and analystics (AusDM'06), Sydney, Australia, pp. 83–89. Australian Computer Society (2006)

Using Information Visualization and Text Mining to Facilitate the Conduction of Systematic Literature Reviews

Sandra Fabbri[1(✉)], Elis Hernandes[1], Andre Di Thommazo[1,2], Anderson Belgamo[1,2,3], Augusto Zamboni[1], and Cleiton Silva[1,2]

[1]LaPES-Software Engineering Research Lab, Universidade Federal de São Carlos, São Carlos, SP, Brazil
{sfabbri, elis_hernandes, augusto_zamboni}@dc.ufscar.br
[2]Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, São Carlos, SP, Brazil
{andredt, anderson, cleitonrqs}@ifsp.edu.br
[3]Methodist University of Piracicaba, Piracicaba, SP, Brazil

Abstract. Systematic Literature Review (SLR or SR) and Systematic Mapping (SM) are scientific literature review techniques that follow well-defined stages, according to a protocol previously elaborated. Besides systematizing the search for relevant studies, the SR predicts the organization and the analysis of the obtained results. However, the SR application is laborious because there are many steps to be followed. Aiming to offer computational support to SR and SM, the StArt (State of the Art through Systematic Review) tool was developed. Besides helping the steps of SR or SM, the StArt tool has implemented visualization and text mining techniques to support the conduction and the reporting of the SR or SM. A comparative analysis was carried out in relation to StArt and other similar tools.

Keywords: Systematic literature review · Systematic mapping · Tool · Visualization

1 Introduction

The Systematic Literature Review process (SR) has its origins in the medical area and its objective, according to Pai et al. [1], is the creation of a complete and impartial summary about a given research topic following well defined and known procedures. Recently, this process is being adapted to the computer science area, particularly in Software Engineering [2]. Some advantages of the SR usage are the coverage, the replicability and the reliability of its process. Besides systematizing the search for relevant studies, the SR predicts the organization and the analysis of the obtained results. However, the SR process is more laborious than the research conducted on an informal basis [2].

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 243–256, 2013.
DOI: 10.1007/978-3-642-40654-6_15,

A previous activity to the SR should be the Systematic Mapping (SM) which objective, according to Petersen et al. [3], is to build a classification scheme and to structure a software engineering research area. Like a SR, SM is also a laborious activity and its process is similar to the SR process, with many repetitive steps. One of the main differences between SR and SM is that the desired results of SMs are mainly quantitative and the selected studies cannot be read in full. Despite this fact, quantitative data can also aid the summarization that should be provided by a SR.

Thus, considering that there are several steps to be executed and several documents to be managed, computer support can aid the conformance to the SR and SM processes, enabling higher quality in their execution.

Since 2006, the StArt tool [4] has been developed. In 2008 it was completely restructured and a new version is available [5, 6]. This version provides full support to carry out SRs and visualization and text mining are being added for easing data summarization since, in general, the SR outcome generate a lot of data for transforming into knowledge. As mentioned by Burley [7], information visualization is a valuable tool for knowledge integration activities and, in StArt, such views allow the researcher to find, in a simple way, information on the most important events, the evolution of the research topic by the academic community, and so on. This information is very common in SM.

Another important contribution that has been reached with information visualization in StArt is the evaluation of the search strings quality. An important point in this kind of literature reviews is to find and ensure that the search strings bring all the relevant studies on the research topic. The StArt tool provides visualization of all the studies retrieved as well as their references. Hence, it is possible identifying, for example, if a frequently cited reference was or was not retrieved by the search string.

Based on this context, the objective of this paper is to explore the contributions of information visualization for these kinds of literature reviews. Section 2 presents an overview of StArt functionalities and highlights some features that aid the control of SR and SM processes. Section 3 explains the visualization support provided by StArt and how it can be used to enhance the summarization of the investigated topic. Section 4 presents the support of text mining processing. Section 5 presents a comparative analysis of related tools and Sect. 6 presents the conclusions and future work.

2 An Overview of the StArt

Before explaining how information visualization and text mining processing help on identifying important information for SM and SRs, an overview of the main functionalities of StArt is presented below. As mentioned before, the processes of SR and SM have some repetitive steps and require discipline and systematic practice from the researcher. The information must be registered in an organized way, such that the expected results are reached, the process can be replicable and all the information can be packed.

Thus, StArt has been developed for providing automated support to as many steps as possible. Functionalities to ease data summarization were also implemented in the tool as the possibility to display data through visualization and Excel formatted reports, according to researcher's needs.

As the SM process is a subset of the SR process, StArt was initially planned to support SRs and currently it is being adapted to also support SMs. Figure 1 illustrates the general process of SR, highlighting what is done with (left side) and without (right side) StArt support. As electronic scientific databases do not allow automated search of primary studies, steps 2, 3 and 4 must be executed without the support of the tool. They are: the adjustment of search strings in search engines, which happens while the protocol is being defined and reviewed; the execution of these search strings after the protocol approval; and the exportation of the search result in a BibTex file, respectively. The step numbers used in this figure will be used in the explanation of the StArt functionalities.

The main functionalities of StArt are presented in the screen shot of Fig. 2. At the left side there is the hierarchical directory tree with the SR process phases. At the right side, the information associated to the functionality selected on the left side is presented.

Shortly, the goals of the three phases are:

- Planning Phase, which consists of the protocol filling (Step 1 of Fig. 1);
- Execution Phase, which is composed of Studies Identification (Steps 2, 3, 4, and 5 of Fig. 1), Selection (Steps 6, 7, and 8 of Fig. 1) and Extraction (Step 9 of Fig. 1). In this phase the researcher should identify the studies, select them and extract the relevant information for answering the research question.
- Summarization Phase (Steps 10 and 11 of Fig. 1), which corresponds to the analysis of the data extracted from each accepted study and the elaboration of a final report describing the state of the art. For this phase, StArt provides graphics, spreadsheets and data visualizations, aiming to make the researcher's tasks easier. Such options will be detailed in Sect. 4.

In the next sections, each phase is discussed in detail, exemplifying the support provided by the StArt tool.

2.1 Planning

In this phase StArt supports the SR Protocol elaboration (Step 1 of Fig. 1) according to the attributes suggested by [8]. Some of the attributes are: research question definition (the research question that the review is intended answer); keywords that will be used for searching for studies; search engines (examples: ACM, IEEE, Scopus); criteria for acceptance or rejection of studies; etc. There is a help message for each protocol attribute aiming to guide its filling. The protocol is stored in the tool and can be accessed and modified if necessary. It is worth noting that, to ensure the SR process conformance, the content of the protocol fields are reflected in later steps of the SR process. For example, when a search engine is chosen during the protocol filling, it is added under the Studies Identification of the Execution Phase, as shown in Fig. 3. Similarly, each attribute inserted in the Information Extraction Form Attributes during

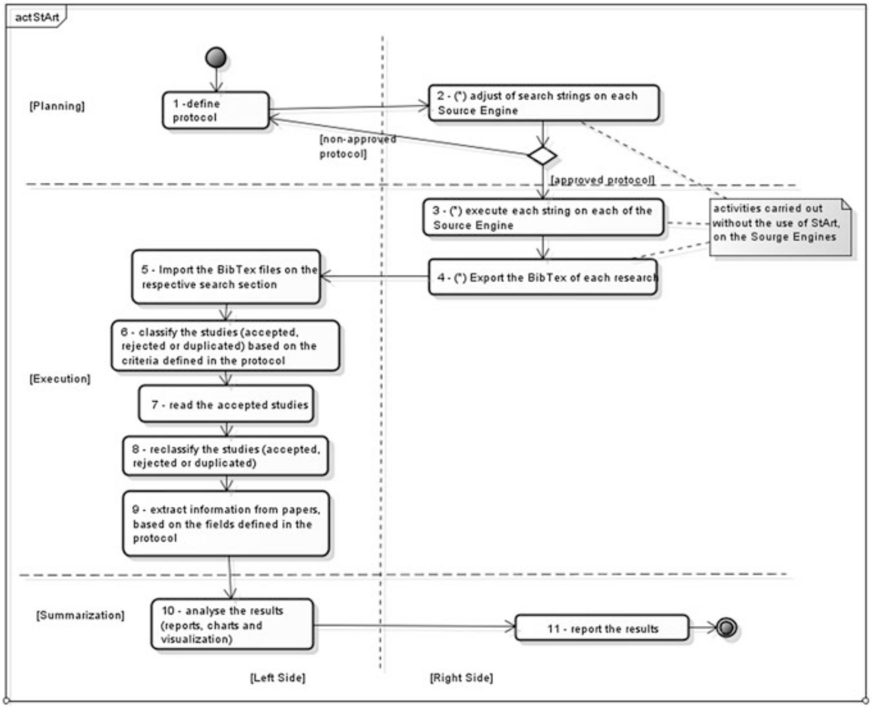


Fig. 1. SLR steps: Left side – actions supported by StArt and Right side – actions not supported by StArt.

the protocol filling becomes a field that must be filled in during the Extraction Step (Step 9 of Fig. 1), as shown in Fig. 4.

2.2 Execution

This phase of the SR has three steps according to the guidelines proposed in [2, 8]. The first one is Studies Identification (Steps 2–5 of Fig. 1). In this step, the researcher should adjust the search string using the keywords earlier defined in the protocol. After this step, the strings should be applied in each search engine, for example, IEEE, Scopus, ACM, Springer and Web of Science. This action is not supported by the tool and the search results must be imported into StArt. As the studies are being imported into the tool, a score is assigned for each one, depending on the keywords defined in the protocol appear in the title, abstract and keywords list of each study. This score can be used, for example, to establish an order of reading once studies with higher scores should be more relevant to the SR. Also, if the studies with higher scores are not relevant to the research question, it is possible that the strings should be revisited and improved. The string definition is an important point to the success of SRs, and its

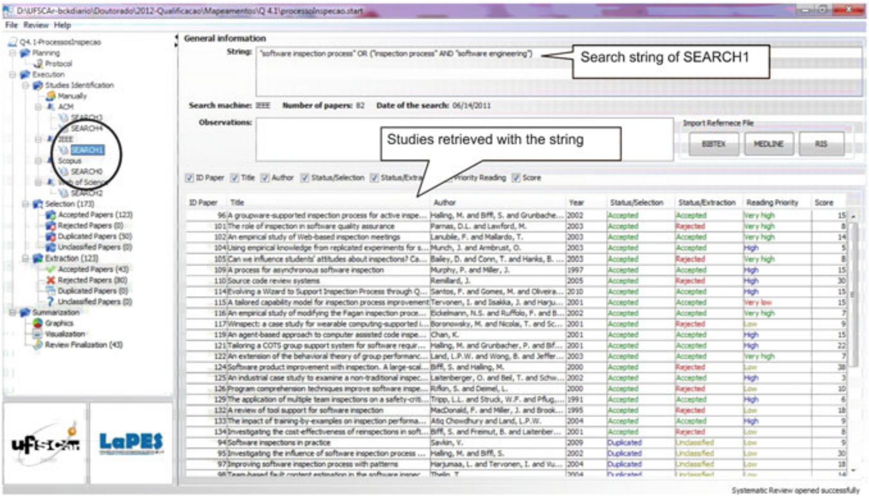


Fig. 2. Overview of the StArt tool.

Fig. 3. Search engines defined in the protocol are automatically inserted under Studies Identification.

quality can be accessed through visualization provide by StArt, which is explored and presented in Sect. 4.

The second step is Studies Selection (Step 6 of Fig. 1). In this step, the researcher should use the inclusion and exclusion criteria, defined in the protocol, to classify the studies as accepted or rejected. Duplicated studies are automatically identified by the tool. When the study is accepted, the researcher can attribute to it a relevance level (Very High, High, Low or Very Low).

The third step is Extraction (Steps 7, 8 and 9 of Fig. 2). At this step, the researcher must read the full version of each "Accepted study", elaborate a summary and fill in its Information Extraction Form (Fig. 4B).

Aiming to facilitate this step, it is possible to link the full text file (e.g. PDF files) of each study with its record in the tool.

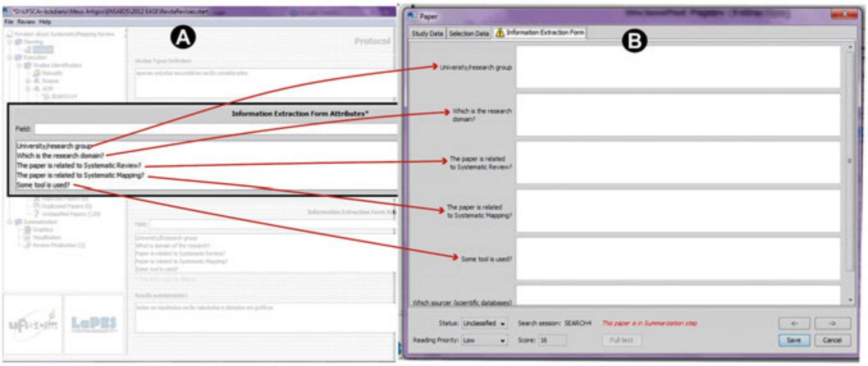


Fig. 4. Relationship between attributes defined in the protocol and the form available during the Extraction step.

2.3 Summarization

In this phase (Step 10 of Fig. 2), StArt provides the following facilities:

- Easy access to the information of all studies accepted in Extraction Step. Comments and information extracted in previous steps can be accessed and copied to a text editor added in the tool. After collecting that information, the researcher can transfer this initial version of the summary to a more powerful text editor.
- Generation of charts that support a quantitative SR characterization. For example: the percentage of studies identified by each search engine, the percentage of studies accepted, rejected and duplicated in Extraction step, the times that each inclusion

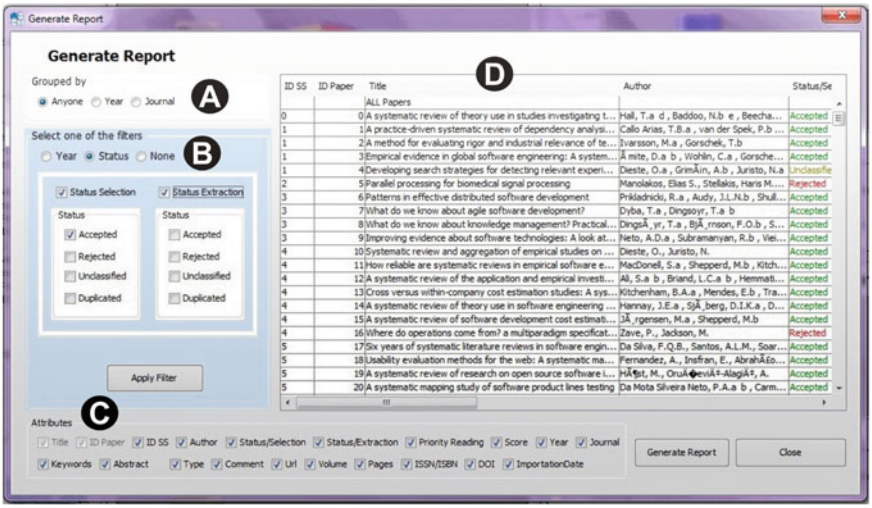


Fig. 5. Options for specifying reports.

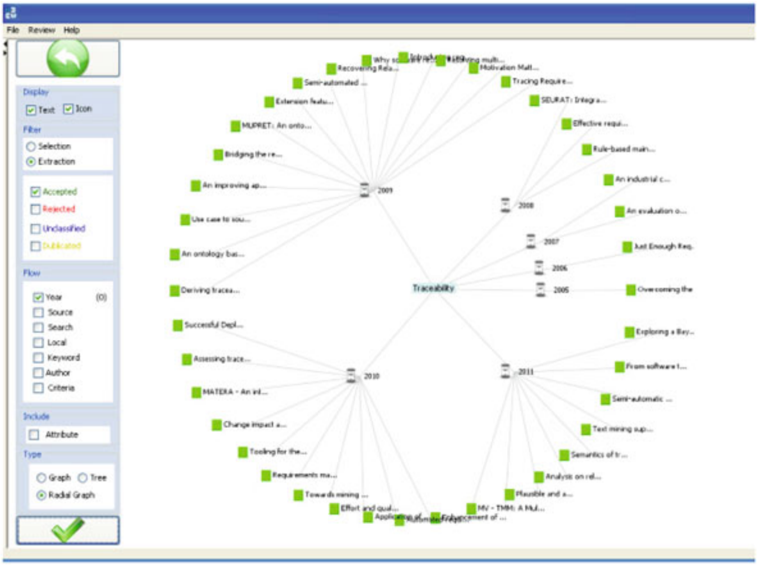


Fig. 6. Visualization of publications on "traceability" over time.

and exclusion criterion was used for classifying the studies as accepted or rejected (Fig. 11). In fact, this kind of quantitative data is particularly relevant for Systematic Mappings [3]. In case the researcher choose to do meta-analysis, carry out statistical tests or elaborate other charts, StArt can generate, among other reports, a spreadsheet that allows data manipulation outside the tool. These reports can be generated according to researchers' needs, based on options that allow grouping data in different ways, (Fig. 5A), applying different filters (Fig. 5B) and choosing specific characteristics of the studies (Fig. 5C). Figure 5D shows a preview of the report.

- Deal with a large volume of data to discover features, patterns and hidden trends through visualization. When an SR or SM process is finished, there is a large amount of data related to the research topic that can show trends in the evolution of the topic over time, which is interesting information to explain the state of the art. As mentioned before, the information visualization is a helpful tool for knowledge integration activities.

3 Visualization in StArt

Considering the importance of quantitative data for both the SR and SM and the fact that information visualization explores the natural visual ability of humans aiming to facilitate information processing [9], StArt uses visualization to facilitate knowledge management about literature reviews. Using effective visual interfaces, it is possible to

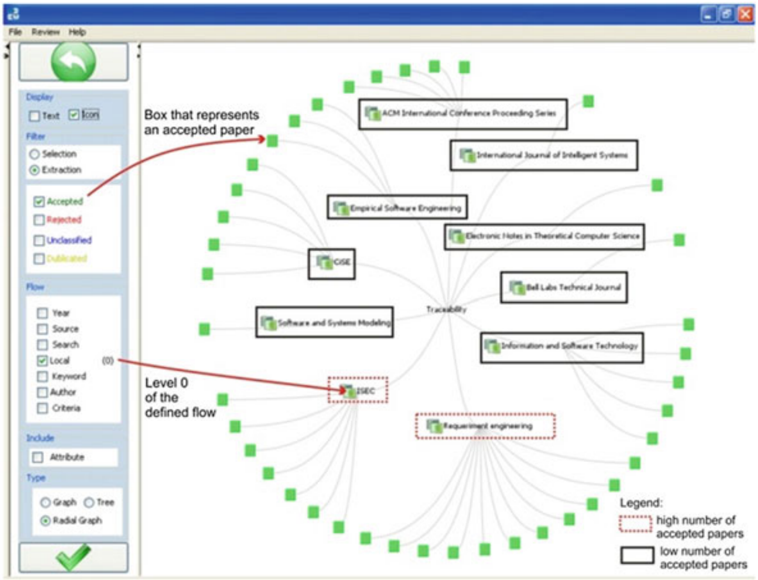


Fig. 7. Visualization of conferences/journals where “traceability” has been published.

quickly manipulate large volumes of data to discover characteristics, patterns and hidden trends.

Based on visualization, for example, it is easier to realize how a specific research topic evolved over time. See Fig. 6 where the researcher’s interest was to understand how the topic “traceability” was explored by the academic community, in relation to the question investigated in this example. It is easy to identify that in 2005 and 2006 there was only one published study; in 2007 and 2008 there were few additional studies, but in 2009, 2010 and 2011, the number of studies that mentioned the research topic was more significant than in the previous years.

To build this visualization, the researcher should select the following options (Fig. 6): green rectangle representing an accepted study; part of the study title nearby the rectangle, the publication year as the grouping filter, and the Radial Graph as the visualization technique.

Now, suppose that the researcher would like to identify appropriated places for submitting a study or for publishing results of a literature review. In this case he/she should select almost the same options mentioned before, exchanging year by place. This visualization (Fig. 7), allows identifying the main discussion forums for the topic under investigation. Observe that some places have few studies related to “traceability”, while some others have more publications on this topic. Besides, the visualization type was Radial Graph and the studies titles were omitted.

If the researcher wishes to merge both the previous analysis in one graph, it would be better to use a different visualization type. In this case the Tree technique seems

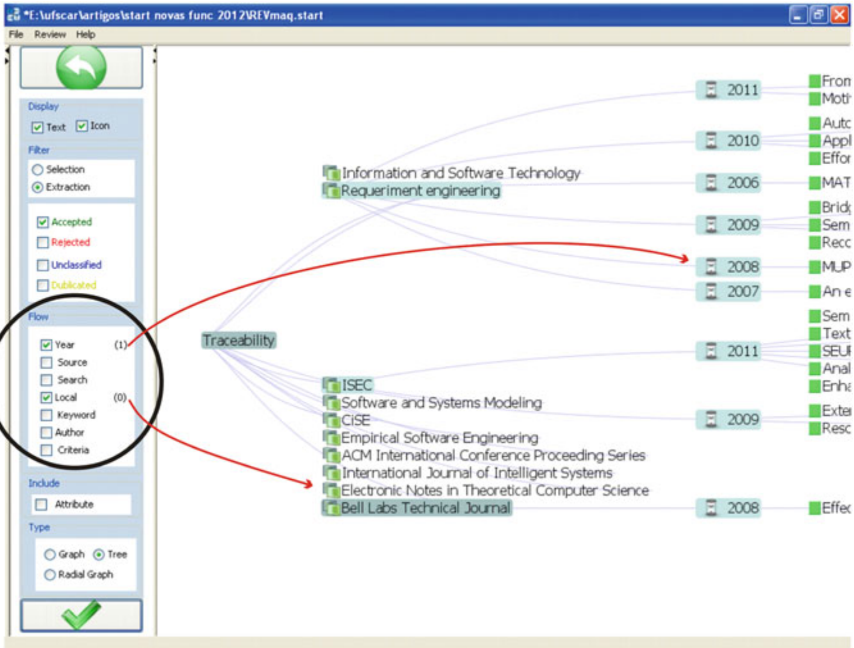


Fig. 8. Visualization of conferences/journals where "traceability" issue has been published according to the year.

better, as shown in the screenshot of Fig. 8. The researcher can expand the levels according to their need.

A double click on a selected study shows information (like authors, abstracts, etc) about it.

In addition to the features described above, visualization is also used to show the relationship among the studies recovered in literature review. This information allows evaluating the set of studies and enhancing the search for them. This resource is better explained in next section.

4 Text Mining in StArt

According to [10], the growing number of publications combined with increasingly cross-disciplinary sources makes it challenging to follow emerging research topics and identify key studies. It is even harder to begin exploring a new field without a starting set of references.

During the conduction of literature reviews many studies are retrieved from various search engines through search strings. Hence, the researcher must be careful not to leave out any studies that may be relevant. According to [11], the usual problem of systematic reviews is that the more inclusive the search strategy, the more irrelevant

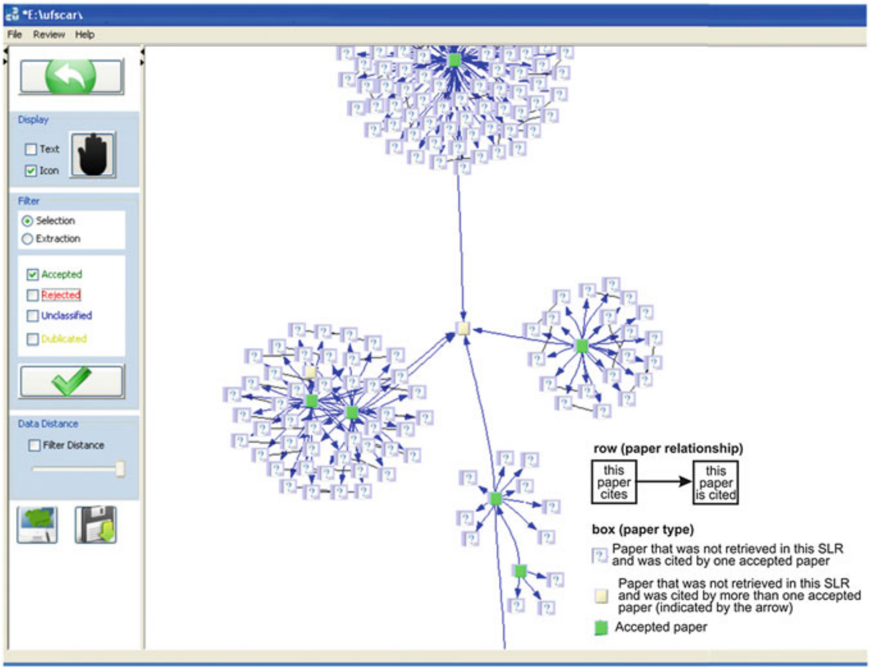


Fig. 9. Relationship among the studies uploaded into StArt and their references.

studies will be retrieved; the more precise and specific the search strategy, the more relevant studies will be missed.

In order to help minimizing this problem, StArt provides support to identify the references of each study retrieved by the search strings. This support allows knowing if there are studies not retrieved, but referenced.

As the search engines generally do not provide the list of references from each study, this information is obtained by reading and extracting the references of the PDF files of the retrieved studies. Every time a PDF file is linked to a study, StArt searches the references in the PDF file. Aiming to identify information like authors, publication place and title, regular expressions are used to identify the bibliographic reference template that was used (APA, Harvard, IEEE, etc.). To determine which study is related to another one, the similarity between the titles of the studies is calculated using the text mining algorithm proposed by [12]. The result of this process is shown through visualization as presented in Fig. 9. The study in the centre of the figure was not retrieved in the literature review, but is referenced by five studies that were retrieved.

This functionality is especially useful during the execution of pilot literature reviews, which should be conducted for adjusting the protocol and the search strings, as suggested by [8]. If there are studies not found but referenced many times, the

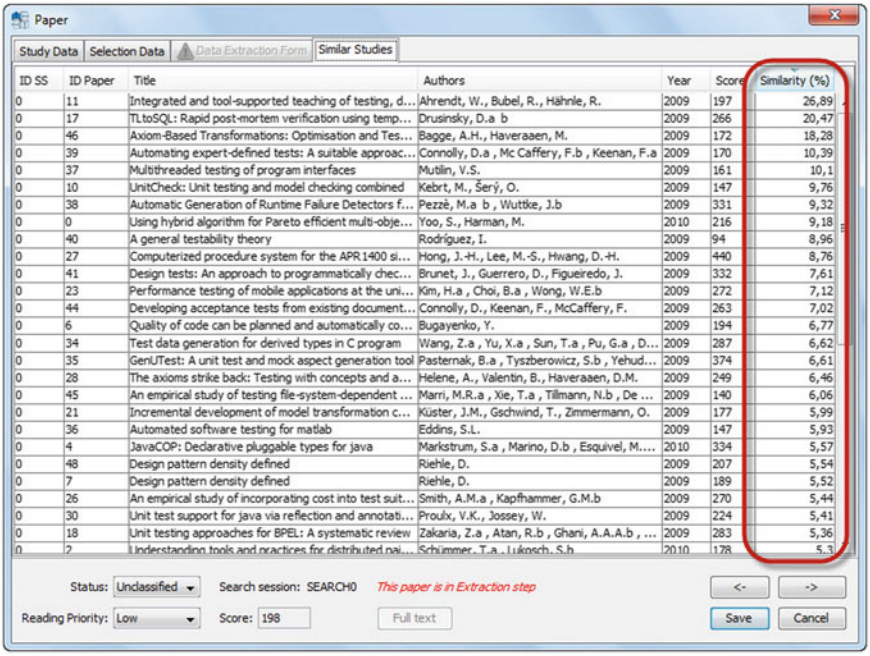


Fig. 10. List of similar studies in relation to a selected study – the similarity grade is highlighted.

researcher should verify, for example, if the keywords of these studies should be considered in the protocol and search strings. If so, a new search applying these new keywords must be performed aiming to find relevant studies that were missed.

Start also offers the functionality for detecting which of the studies imported into the tool are similar. The similarity is calculated based on the abstracts through Vector Processing Model [13]. The result of this processing is shown in a table as presented in Fig. 10. This table provides a list of similar studies and their respective similarity grade in relation to a study previously selected.

This list of similar studies can be used, for example: (i) to define the next study to be analyzed; (ii) to facilitate comparison between similar studies and (iii) to make the inclusion and exclusion of studies easier – studies with a high level of similarity to an excluded study tend to be also excluded.

Other researches use text mining in the context of SR or SM, but it is not available in tools that support the whole SR or SM processes.

Malheiros et al. [14] proposed the use of a visualization tool, named PEx, to support the first step of studies selection. PEx has a module that processes the abstract of the primary studies, eliminates stopwords, calculates the terms frequency and, based on this result, displays clusters of studies to facilitate their analysis.

Felizardo [15] continued the previous research and presented the VTM (Visual Text Mining) tool which supports studies selection. As proposed by Malheiros [14], the result of text mining processes is shown by different visualization techniques which help applying the inclusion and exclusion criteria previously inserted in VTM tool.

It is important to notice that the focus of these studies is the studies selection step. On the other hand, in Start, visualization and text mining are currently being used to support the search string definition and the SR or SM Summarization phase.

5 Related Tools

In the literature, there are some tools to support the management of bibliographic references, which are commonly used by researchers to aid in the SLR process. The purpose and the coverage of these tools are different and they are not related to the SLR process proposed by Kitchenham [2], except for SLR Tool [16].

Only SLR Tool [16] focuses on Systematic Literature Review. However, its installation requires the availability of a specific database management system and a pre-configuration of the environment, which can restrict its use, mainly by researchers of other research areas such as Medicine and Nursing, who are also supporters of the SLR process.

Another characteristic of SLR Tool is that it only works with the English and the Spanish versions of the Windows operating system. StArt, on the other hand, can be easily installed, since it has a simplified installer and it can also run with a copy of the executable on the researchers' machines. Table 1 presents related tools that were analyzed and shows some of their features.

Table 1. Features of the StArt and the tools found in the literature.

	Free	Protocol definition	Bibliographic reference manager	Citation export	Attributes customization	Automatic classification of papers	Similar papers identification
JabRef	Yes	No	Yes	Yes	Yes	No	No
EndNote	No	No	Yes	Yes	Yes	No	No
ProCite	No	No	Yes	No	No	No	No
Reference manager	No	No	Yes	No	No	No	No
RefWorks	No	No	Yes	Yes	No	No	No
BibEdt	Yes	No	Yes	No	No	No	No
Zotero	Yes	No	Yes	Yes	Yes	No	No
Biblioscape	No	No	Yes	No	No	No	No
Bookends	Yes	No	Yes	Yes	Yes	No	No
Library master	No	No	Yes	Yes	Yes	No	No
Mendeley	No	No	Yes	Yes	Yes	No	No
Mekentosj	No	No	Yes	Yes	No	No	No
SLR tool	Yes	Yes	Yes	Yes	No	No	No
StArt	Yes	Yes	Yes	No	Yes	Yes	Yes

6 Conclusions

This paper explored the use of visualization for making easier the interpretation of data provided by Systematic Literature Review and Systematic Mapping. This visualization is available in StArt, which also supports the steps of SR and SM processes. As these processes are laborious, posses many repetitive steps and require that all information is packed, the availability of computational support is relevant.

Although there are some tools that have been used by researches to aid the conduction of literature reviews, most of them are reference manager. Some examples are JabRef (jabref.sourceforge.net), EndNote (www.endnote.com), ProCite (www.procite.com), Reference Manager (www.refman.com), RefWorks (www.refworks.com) and Zotero (www.zotero.org). Only SLR tool [16] focuses on SR process [8]. However, it works only on the English or Spanish versions of the Windows operating system.

As StArt is closely associated to the SR and SM processes, it provides many facilities that make easier the conduction of these types of reviews. Some characteristics that differentiate it from the other tools are the score, which is calculated automatically and can give insights on the paper relevance; different types of data visualization that can aid to map the research area; extraction of the references of the studies gathered in the review, that allows evaluating the adequacy of search strings and improving the quality of the whole activity; and other facilities that make the conduction of the process more manageable.

Considering the importance of packing the SRs or SMs data, StArt saves all data in a ".start" file which allows conducting a review in sessions and sharing a review with another researcher. In addition, as StArt provides a simple text editor for writing an initial summary of the state of the art, this summary is also packed. StArt is being continuously evolved and tested. The tool was also evaluated from the perspective of its usefulness and ease of use, according to the TAM model, which found that the tool is useful to users and can be easily used by researchers [6].

As future work, it is planned to continue the development of StArt emphasizing the analysis related to Systematic Mappings. This objective has already initiated with the addition of visualization, but there are other features that can enhance its support for SM. Besides, it is planned some experimental studies that aim to establish a strategy to improve search strings based on the references of the collected studies and also to explore the tool as a support to conduct meta reviews.

Acknowledgements. The authors thank the students and researchers who have been used StArt and are giving constant feedback to development team and CNPq, CAPES and *Observatório da Educação* Project for financial support.

References

1. Pai, M., McCulloch, M., Gorman, J.D., Pai, N., Enanoria, W., Kennedy, G., Tharyan, P., Colford Jr, J.M.: Clinical research methods - systematic reviews and meta-analyses: an illustrated, step-by-step guide. Natl Med. J. India **17**, 89–95 (2004)

2. Kitchenham, B.A.: Procedures for performing systematic reviews. Software Engineering Group, Keele University, Keele, Tech. Rep. TR/SE 0401 (2004)
3. Petersen, K., et al.: Systematic mapping studies in software engineering. In: Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, Bari, Italy (2008)
4. Montebelo, R.P., et al.: SRAT (Systematic Review Automatic Tool) Uma Ferramenta Computacional de Apoio à Revisão Sistemática. In: V Experimental Software Engineering Latin American Workshop, ICMC-São Carlos (2007)
5. Zamboni, A.B., Thommazo, A.D., Hernandes, E.C.M., Fabbri, S.C.P.F.: StArt Uma Ferramenta Computacional de Apoio à Revisão Sistemática. In: Brazilian Conference on Software: Theory and Practice - Tools session, UFBA (2010)
6. Hernandes, E.C.M., Zamboni, A.B., Thommazo, A.D., Fabbri, S.C.P.F.: Avaliação da ferramenta StArt utilizando o modelo TAM e o paradigma GQM. In: X Experimental Software Engineering Latin American Workshop, ICMC-São Carlos (2010)
7. Burley, D.: Information visualization as a knowledge integration tool. J. Knowl. Manage. Pract. **11** (2010)
8. Kitchenham, B.A.: Guidelines for performing systematic literature reviews in software. Software Engineering Group, Keele University, Keele, University of Durham, Durham, Tech. Rep. EBSE-2007-01 (2007)
9. Gershon, N., Eick, S.G., Card, S.: Information visualization interactions. ACM Interact. **5**, 9–15 (1998) (ACM Press)
10. Dunne, C., Shneiderman, B., Gove, R., Klavans, J., Dorr, B.: Rapid understanding of scientific paper collections: integrating statistics, text analytics, and visualization. JASIST: J. Am. Soc. Inf. Sci. Technol. **63**, 2351–2369 (2012)
11. Boell, S.K., Cezec-Kecmanovic, D.: Are systematic reviews better, less biased and of higher quality? In: Proceedings of the European Conference on Information Systems, Helsinki, Finland (2010)
12. Salton, G.: Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)
13. Salton, G., Allan, J.: Text retrieval using the vector processing model. In: Symposium on Document Analysis and Information Retrieval. University of Nevada, Las Vegas (1994)
14. Malheiros, V., Höhn, E., Pinho, R., Mendonça, M., Maldonado, J.C.: A visual text mining approach for systematic reviews. In: International Symposium on Empirical Software Engineering and Measurement, ESEM, pp. 245–254 (2007)
15. Felizardo, K.R., et al.: Using visual text mining to support the study selection activity in systematic literature reviews. In: International Symposium on Empirical Software Engineering and Measurement, ESEM, pp. 77–86 (2011)
16. Fernández-Sáez, A.M., Genero, M., Romero, F.P.: SLR-Tool: a tool for performing systematic literature reviews. In: Proceedings of the JISBD, pp. 329–332 (2010)

Accelerating Crosscutting Framework Reuse Using a Model-Based Approach

Thiago Gottardi[1,(✉)], Oscar Pastor López[2], and Valter Vieira de Camargo[1]

[1] Departamento de Computação, Universidade Federal de São Carlos, Caixa Postal 676, São Carlos–SP 13565-905, Brazil
{thiago_gottardi, valter}@dc.ufscar.br
[2] Universidad Politecnica de Valencia, Camino de Vera, s/n, Valencia, Spain
opastor@dsic.upv.es

Abstract. Infrastructure frameworks are used to accelerate the development of information systems. These frameworks encompass software concerns commonly used in the information system domain, e.g. "persistence", "authentication" and "concurrency". Although these functionalities improve productivity, the reuse process is still heavily based on writing source code, which forces the development teams to learn low level details of the frameworks being reused, causing the reuse process to be entirely performed during implementation phases. In this article, the focus is on Crosscutting Frameworks, which modularize specific types of software concerns. Since these frameworks can also be infrastructure frameworks, in this article, a new model-driven process is applied in a persistence crosscutting framework. A tool was created to support the process, which was successfully evaluated in an empirical study. This process was compared to the conventional process in an empirical study in order to evaluate the efforts needed to successfully reuse the framework. In our study, the tool usage has reduced the reuse process time by more than 97 %.

Keywords: Model-driven engineering · Framework reuse · Aspect-oriented programming · Crosscutting framework · Empirical study

1 Introduction

Enterprise Information Systems (EIS) are fundamental for many companies, which makes important to be able to create more quality EIS in less time. Reusing software frameworks can be used to deal with this issue. Frameworks were firstly defined by Fayad [1] as "sets of reusable and customizable software components for specific application domains". There are many types of frameworks, but in this paper, we concentrate on a specific kind of framework called "Crosscutting Framework" (CF), which are aspect-oriented frameworks.

Aspect-Oriented Programming (AOP) was created to improve the modularization of a system by providing language abstractions for crosscutting concerns, which could not be well modularized using previous paradigms [2]. As soon as

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 257–273, 2013.
DOI: 10.1007/978-3-642-40654-6_16,

the first Aspect-Oriented languages became available, researchers proposed new techniques to improve reuse of crosscutting concerns, among those proposals are "Crosscutting Frameworks" (CF). Crosscutting Frameworks are intended to modularize and ease reuse of a single crosscutting concern that may affect a software system, for example, persistence, concurrency, authentication and business rules. Also, these frameworks can be customized to better fit into the software requirements [3].

The conventional reuse process of most CFs found in literature apply white-box reuse strategies in their instantiation process, relying on writing source code to reuse the framework [3–14]. This abstraction level forces application engineers to worry about low level details of implementation, which leads to the following problems: the application engineer must know coding details regarding the programming paradigm employed to develop the framework, which makes the CF reuse process learning curve steeper; coding mistakes are more likely to happen when the reuse code is created manually; several lines of code must be written for the definition of small number of information needed during the reuse process, impacting development productivity; reuse process can only be started during implementation phase and the framework reuse documentation, e.g., "cookbook", may be complex to understand.

In this paper, a new Model Driven Development (MDD) approach is presented. Its main objective is to shorten development time of applications that reuse a specific type of framework. MDD combines generative programming, domain specific languages and software model transformation. Its objective is to shorten the gap between the problem and solution, by applying models that protect developers from implementation platform complexity [15]. These languages are used to express domain concepts in a more effective way, while transformations are performed to convert models to codes or other models [16,17]. Therefore, we employ MDD on the purpose of raising the abstraction levels of reuse process.

The model must be created by the framework developer after developing a framework. The idea is to release the model along with the framework code to support its reuse by replacing textual cookbooks. This model should then be used by an application engineer in order to support the reuse of the framework.

The use of the model is inserted in a new process for CF reuse, which allows an application engineer to reuse these frameworks in our model driven approach.

This paper is structured as follows: in Sect. 2, there are related works, in Sect. 2.1, Crosscutting Frameworks are explained; in Sect. 3, the Proposed Model and the Reuse Process are shown; in Sect. 4, a tool to support the process is used to reuse a persistence framework as an example; in Sect. 5, an empirical evaluation is presented and in Sect. 6, there are the conclusions.

2 Related Works

The approach proposed by Cechticky et al. [23] allows object-oriented application framework reuse by using a tool called OBS Instantiation Environment. That tool supports graphical models do define the settings of the expected application to be generated. The model to code transformation generates a new application that reuses the framework.

The proposal found in this paper differs from their approach on the following topics: (1) their approach is restricted to frameworks known during the development of the tool; (2) it does not use aspect orientation; (3) the reuse process is applied on application frameworks, which are used to create new applications.

Another approach was proposed by Oliveira et al. [24,25]. Their approach can be applied to a greater number of object oriented frameworks. After the framework development, the framework developer may use the approach to ease the reuse by writing the cookbook in a formal language known as Reuse Definition Language (RDL) which also can be used to generate the source code. This process allows to select the variabilities and resources during reuse, as long as the framework engineer specifies the RDL code correctly.

These approaches were created to support the reuse during the final development stages. Therefore, the approach proposed in this paper differs from others by the supporting earlier development phases. This allows the application engineer to initiate the reuse process since the analysis phase while developing an application compatible to the reused frameworks. Although the approach proposed by Cechticky et al. [23] is specific for only one framework, its can be employed since the design phase. The other related approach can be employed in a higher number of frameworks, however it is used in a lower abstraction level, and does not support the design phase. Other difference is the generation of AOP, which improves code modularization.

2.1 Aspect-Orientation and Crosscutting Frameworks

Aspect-Oriented Programming (AOP) is a paradigm created to improve code modularization. AOP languages provide constructions to allow modularization of crosscutting concerns, which are concerns that may affect several parts of code and cannot be modularized correctly with many other paradigms, e.g. Object-Oriented [18]. Among these constructions, Pointcuts are used to capture join-points of an application, which may receive new behavior when affected by a crosscutting concern. Examples of join-points include method executions, object constructions and attribute definitions. Advices are constructions similar to operations that define a behavior to be applied on join-points captured by Pointcuts. For example, an advice may be defined to apply a behavior on several join-points to reduce code replication. Aspects are constructions similar to classes used to modularize Pointcuts and Advices.

Crosscutting Frameworks encapsulate the generic behavior of a single crosscutting concern [3]. There are crosscutting frameworks developed for persistence [3,6], security [5], cryptography [8], distribution [6] and other concerns [4]. Their main objective is to make to reuse of such concerns easier during the development of an application without the need to use explicit function calls from the base code.

As well as other types of frameworks, Crosscutting Frameworks also need information regarding the base application in order to be coupled correctly. For instance, reuse requirements for an access control CF may be: (1) informing methods that should receive access control; (2) informing user role names for system users; (3) informing how many times a user is allowed to enter an incorrect password before being blocked.

Unlike application frameworks, which are used to generate a whole new application, a CF needs to be coupled to a base application in order to become functional. The standard usual process to reuse a CF is composed by two activities: instantiation and composition. Instantiation is the conventional reuse process in which framework code is specialized for the base application. During this process, classes are extended, hooks are defined, variabilities are chosen or implemented. During the composition activity, pointcuts and composition rules are defined, unifying the chosen variabilities and the base code. The code created specifically to reuse an CF is referred as "reuse code".

The final application is composed by three types of code modules: base, reuse and framework. The "base code" represents code of the base application. In the "framework code" there is the code of the CF, which is untouched during the reuse process. The "reuse module" is the connection between the base application and a framework. Each final application can be composed by several frameworks, each one coupled by a reuse module. However, there must be only one base module, which encompasses the main method, also known as the application entry point.

3 Model-Driven Approach for CF Reuse

3.1 Proposed Model

In this paper, we propose a new model named "Reuse Model" (RM). We also created a graphical form to represent the model as shown on Fig. 1.

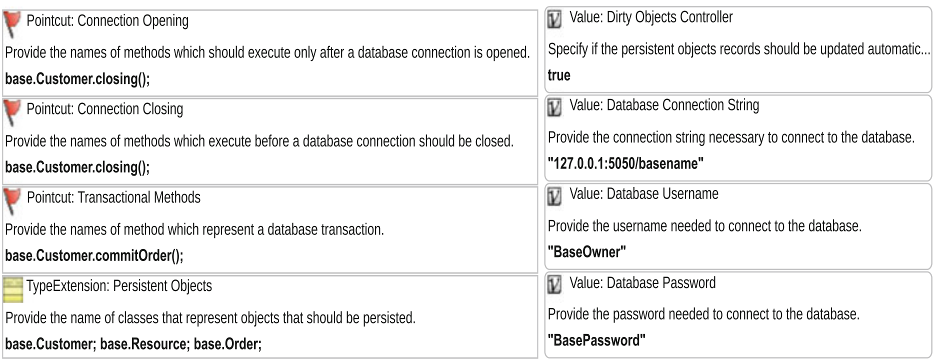


Fig. 1. Reuse Model Form.

The RM should be provided by the framework developer in order to represent a reuse documentation in a high level. Each CF available for reuse should have its own RM. The idea is to release this model along with the framework to support its reuse process. Since the framework developer has good knowledge regarding the framework, that person is able to document how the reuse should be performed. The RM is also used to document the information needed by the framework in order to be coupled to the base code.

The RM is intended to improve the understandability of the framework during reuse process. By analyzing the model, an engineer reusing the framework should be able to learn about the information needed during the reuse process. This model also represents the variabilities provided by a framework that must be chosen by that engineer.

There are five possible elements in the presented model: “Pointcuts”, “Type Extensions”, “Options”, “Option Groups” and “Values”. The “PointCuts” represent join-points of the base application code that should be affected by the crosscutting framework; “Type Extensions” represent types found in the base application that must extend or implement classes, aspects or interfaces found in the crosscutting framework. “Option” and “Option Group” represent variabilities provided by the CF that may be chosen by the application engineer and “Value” represent any other numeric or textual values that must be informed while reusing the framework. For instance, to be able to instantiate a persistence CF, the application engineer must specify methods from base application that should be executed after a database connection is opened and before it is closed. It is also needed to specify methods that represent data base transactions, and the variabilities must be chosen, e.g., the driver which should be used to connect to the database system.

In order to instantiate the framework, the RM may indicate the need of informing join-points of the base code where crosscutting behavior would be applied, as well as classes, interfaces or aspect names that would be affected. Framework variabilities that must be chosen during reuse process are also visible.

The RM is employed to support the reuse process of a crosscutting framework. It is intended that the reuse process can be completely executed by completing the form. Therefore, it should be used by the application developer in order to reuse a framework. When concluding the reuse process by completing the model with the information needed by the framework, the reuse code generation is possible. For example, on Fig. 1 there is an instantiation of the RM for a Persistence CF [19]. By processing the model on a model to code transformation process, the reuse code will be generated. This example and the code generation are further explained in the “Approach Usage Example” section.

3.2 Development Process

A new reuse process is specified when considering the model to support the reuse of Crosscutting Frameworks. In order to explain the new process, there is an activity diagram on Fig. 2 which illustrates the perspective of engineers reusing the framework.

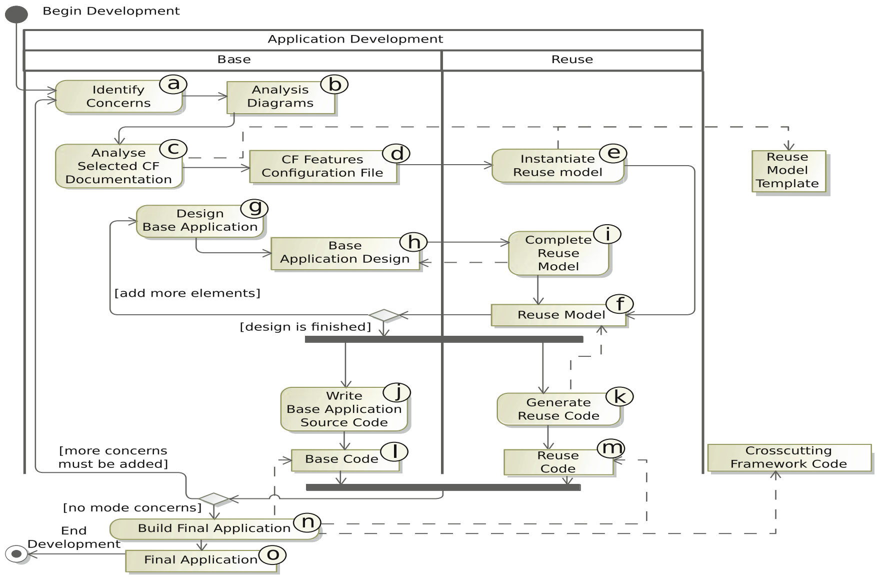


Fig. 2. Reuse Process Activity Diagram.

The reuse process starts on the left side of the figure. The application being developed is composed by the "Base" and "Reuse" modules. By analyzing the application being developed ('a'), the engineer should be able to identify the concerns that would affect the software, possibly by using an analysis diagram ('b'). By identifying these concerns, the engineer has the opportunity to select crosscutting frameworks at that moment and begin the reuse process since earlier development phases ('c'). A configuration file ('d') can be employed to select a subset of the features and variabilities, which are present in some frameworks. This configuration file is also employed to instantiate a reuse model from a template by selecting the groups which represent the selected features ('e').

As established in the process, we recommend to design the application after identifying the frameworks in order to develop a base application design compatible to the framework. After designing a base application ('g')('h'), names of classes and methods are defined, which are enough to complete the reuse model ('i')('f'). Once the design and) the reuse code are complete, it is possible to write the application source code ('j')('l') following the established design and generate the reuse code ('k')('m'). Once all code modules are complete (base, reuse and framework), the final application can be built ('n')('o'), which completes the process.

It is important to point that the tool being presented on this article is part of a larger project we are working in order to create a integrated development environment for applications that reuse crosscutting concerns. This environment also supports selection of subsets of features of a framework and allows application developers to download the framework instance optimized to their needs. These details are not described in this article.

The Reuse Model contains information needed by the framework being reused. By identifying that information during earlier development phases of the base applications, it is easier to define it correctly in a specific way which all information needed can be easily extracted. Consequently, the base application is not oblivious about the framework and its behaviors, however, the modules are completely isolated and have no code dependency among them.

It is important to point that the Reuse Code itself depends on the Base Code during the build process, however, its definition can be made as soon as the base application design is complete.

The advantage of modularizing the reuse code with aspect oriented concepts removes the dependency between the base application and the reuse code. This allows repeating the code generation without affecting the base code. A new tool to support the process is presented in the next section.

4 Approach Usage Example

In this work, a tool was implemented to support the proposed process. The tool was developed using Eclipse Modeling Framework [20]. It is capable of presenting the proposed model as a form and is also able to transform the model to generate reuse code. For example, we employed the tool to perform the reuse process on a Crosscutting Framework that modularizes the persistence concern [19].

The Reuse Model of the CF is shown in Fig. 1. The last line of each box is initially blank, and must be filled by the application developer with details regarding the base application. The pointcuts "Connection Opening", "Connection Closing" and "Transactional Methods" are intended to capture specific join-points of the base application, e.g. methods of the base application that will be affected by the framework. The first two represent, respectively, method that should execute after a database connection is open or before it is closed, whereas the last pointcut represents methods that encapsulate data transactions.

The "Persistent Objects" is a type extension definition, then, it may represent either a class or an interface that should be extended or implemented by a base class or interface. In this case, the application engineer must supply names of classes (or their super-types) which represent objects that should be persisted on the database. The other elements represent framework variabilities that should be defined by the application developer. For example, the form in Fig. 1 is completed with information of a base application. There are three referenced "Persistent Objects"; their classes will receive methods and crosscutting behavior in order to implement the persistence concern and persist their instances in a database.

There are four selected values on the right of the figure. The first one is used to define if the objects should be saved automatically upon modifying their attributes, which can be performed by the "Dirty Objects Controller". The second is employed to define the connection string, finally, the other two inform the connection user-name and password.

After completing the Reuse Model Form, it is possible to execute a code generator, which is a model to code transformation tool capable of generating

```
public aspect ConnectionCompositionReuse
 extends ConnectionComposition {
    public pointcut openConnection():
        execution (* base.Customer.initial());
    public pointcut closeConnection():
        execution (* base.Customer.closing());
    public pointcut transactional():
        execution (* base.Customer.commitOrder());
}
```

```
public aspect OORelationalMappingReuse
 extends OORelationalMapping {
    declare parents: base.Customer
      implements PersistentRoot;
    declare parents: base.Resource
      implements PersistentRoot;
    declare parents: base.Order
      implements PersistentRoot;
}
```

```
public aspect ConnectionVariabilities{
    public String SelectedManager.setDSN(){
        return "127.0.0.1:5050/basename";
    }
    public String SelectedManager.setUsername(){
        return "BaseOwner";
    }
    public String SelectedManager.setPassword(){
        return "BasePassword";
    }
}
```

Fig. 3. Reuse Code

a reuse code in AspectJ, illustrated on Fig. 3, which allows coupling the base application to the framework in a separate module. The final software is the composition of base application code, reuse code for each reused framework and the code of reused frameworks.

In the first aspect, the three pointcuts are implemented by extending an abstract aspect of the framework with information of the methods found in the base application. In the second aspect, the type extension is implemented, then the classes “Customer”, “Resource” and “Order” receive an interface of the framework, which is used to apply crosscutting behavior. In the third aspect, the values are set by overriding methods of the framework. The interface “Selected-Manager” is implemented by classes which extend the selected database connection. However it is not visible in this article, due to size limitations.

5 Evaluation

We conducted empirical studies in order to analyze differences between using the reuse support tool and using the conventional reuse technique, therefore, our goal was to identify which technique takes less effort to reuse a crosscutting framework. The planning of this study was made considering the guidelines proposed by Wohlin et al. [21].

Table 1. Hypotheses.

H0	There is no difference between using our tool and using an ad-hoc reuse process in terms of productivity (time) to successfully couple a CF with an application. Then, the techniques are equivalent. Tc - $Tm \approx 0$
Hp	There is a positive difference between using our tool and using an ad-hoc reuse process in terms of productivity (time) to successfully couple a CF with an application. Then, the conventional technique takes more time than the model based tool. Tc - $Tm > 0$
Hn	There is a negative difference between using our tool and using an ad-hoc reuse process in terms of productivity (time) to successfully couple a CF with an application. Then, the conventional technique takes less time than the model based tool. Tc - $Tm < 0$

5.1 Study Definition

The objective of the study is to compare the efforts regarding the reuse of frameworks by using conventional technique and the model based tool.

A Crosscutting Framework is considered in two reuse techniques: The conventional reuse technique and the model based tool that we created. The quantitative focus is to compare the efforts needed to reuse a framework with the model based reuse tool and the conventional technique. The recorded timings are considered to determine the effort. The qualitative focus is to determine which technique takes less effort during reuse. The experiment was conducted from the perspective of application engineers who intend to reuse CFs. The study object is the ‘effort’ to perform a CF reuse process. The experiments were planned to compare which technique takes less effort during reuse. The subjects were required to reuse frameworks using the different techniques. An information system was created in order to gather the timings. We added code to the reused applications to submit the data to a server which combined all timings data into a database with milliseconds precision. This submission was transparent to the participants.

This study was conducted with students of Computer Science, in this section, they are referred as participants. Sixteen participants took part on the experiment, eight of these were undergraduate students and the other eight were post graduate students. Every participant had prior AspectJ experience and was required to reuse a CF for the persistence concern [19], coupling it to a provided application using either the model support tool or the conventional ad-hoc technique.

The Table 1 contains our formulated hypotheses. There are two variables shown on the table: “Tc” and “Tm”. “Tc” represents the overall time to reuse the framework using the conventional ad-hoc technique while “Tm” represents the overall time to reuse the framework using the model based tool. There are three hypotheses shown on the table: “H0”, “Hp” and “Hn”. The “H0” hypothesis is true when both techniques are equivalent; then, the time spent using the conventional technique minus the time spent using the model-based tool is

Table 2. Study Design.

Phase	Group 1	Group 2
Training	Reuse techniques training	
	Repair shop	
1^{st} reuse	Conventional	Models
Pilot phase	Hotel application	
2^{nd} reuse	Models	Conventional
Pilot phase	Library application	
1^{st} Primary	Conventional	Models
Reuse phase	Deliveries application	
2^{nd} Primary	Models	Conventional
Reuse phase	Flights application	
1^{st} Secondary	Conventional	Models
Reuse phase	Medical clinic application	
2^{nd} Secondary	Models	Conventional
Reuse phase	Restaurant application	

approximately zero. The "Hp" hypothesis is true when the conventional technique takes longer than the model-based tool; then, the time spent to use the conventional technique minus the time of the model-based tool is positive. The "Hn" hypothesis is true when the conventional technique takes longer than the model-based tool; then, the time taken to use the conventional technique minus the time taken to use the model-based tool is negative. As these hypotheses consider different ranges of a single resulting real value, then, they are mutually exclusive and exactly one of them is true.

The dependent variables are those which we analyze in this work. For each study, we provide analysis of the "time spent to complete the process". The independent variables are controlled and manipulated, for example, "Base Application", "Technique" and "Execution Types".

The participants were selected through non probabilistic approach by convenience, i. e., the probability of all population elements belong to the same sample is unknown. They were divided into two groups. Each group was composed by four post graduate students and four undergraduate students. Each group was also balanced considering a characterization form and their results from the pilot study. On Table 2, there are the phases planned for the study.

Base applications were provided along with two documents. The first document is a manual regarding the current reuse technique, and the second document is a list of details, which describes the classes, methods and values regarding the application to be coupled which are needed when reusing the framework. The applications had the same reuse complexity, then, in order to reuse each application, the participants had to specify four values, twelve methods and six classes. Each phase row of the Table 2 is divided into the name of the application and the technique employed to reuse the framework.

5.2 Operation

At first, every student was introduced to the tool and was taught how to reuse the crosscutting framework using the tool and conventionally. During each phase, the students were required to reuse the CF with a provided application. During the following phase, the participants were required to use the opposite technique to reuse an equivalent application.

Initially, the participants signed the consent form and then answered a characterization form, which had questions regarding knowledge about AspectJ constructs, Eclipse IDE and Crosscutting Frameworks.

After concluding the characterization forms, participants were trained on how to reuse the supplied CF by using the model based reuse tool and then conventionally. It is important to note that every participant already had a basic experience with AspectJ and the conventional reuse of crosscutting frameworks.

The pilot experiment was executed after the training. The participants were split into two groups considering the results of characterization forms. The pilot experiment was intended to simulate the Primary Study, except that the applications were different, but equivalent. During the pilot experiment, the participants were allowed to ask questions about any issues they did not understand during the training. This could affect the validity, then, the data from this activity was only used to rebalance the groups.

During the Primary Study, the participants reused other two applications starting with a different technique for each group. The Secondary Study was another experiment with another two applications.

The recorded timings during the reuse processes with both techniques during both study executions are on Table 3. There are five columns in each of these tables, "G." stands for the group of the participant during the activity; "Application" stands for the application being reused; "T." stands for the reuse technique which is either "C" for conventional or "M" for model based tool; "P." column lists an identifying code of the participants (students), whereas the least eight values are allocated to post-graduate students and the rest are undergraduate students; "Time" column lists the time the participant spent to complete each reuse phase.

The information system was able to gather the timings with milliseconds precision considering both the server and clients system clocks. However, used in the calculations only consider the server time, then, the delay of transmission by the computers are not considered, which are believed to be insignificant in this case, because preliminary calculations considering the client clocks did not change the order of results. The raw data captured from the experiments can be also downloaded as spreadsheets from http://www2.dc.ufscar.br/valter/.

5.3 Data Analysis and Interpretation

The timings data gathered during the study are represented graphically in a bar graph, which is plotted on Fig. 4. The same code for each participant and the

Table 3. Reuse Process Timings.

Primary study					Secondary study				
G.	Application	T.	P.	Time	G.	Application	T.	P.	Time
1	Flights	M	15	04:19.952015	2	Clinic	M	10	02:59.467569
1	Flights	M	13	04:58.604963	1	Restaurant	M	13	03:56.785359
1	Flights	M	8	05:18.346829	1	Restaurant	M	15	04:23.629206
2	Deliveries	M	11	05:24.249952	2	Clinic	M	11	04:25.196135
2	Deliveries	M	5	05:31.653952	1	Restaurant	M	8	04:33.954349
2	Deliveries	M	9	05:45.484577	2	Clinic	M	9	04:41.254920
2	Deliveries	M	3	06:16.392424	1	Restaurant	M	12	05:05.524264
2	Deliveries	M	10	06:45.968790	2	Clinic	M	3	05:45.333167
2	Deliveries	M	14	07:05.858718	2	Clinic	M	14	05:57.009310
2	Deliveries	M	6	07:39.300214	2	Clinic	M	5	06:31.365498
2	Deliveries	M	2	08:02.570996	2	Clinic	M	2	06:59.967490
1	Flights	M	1	08:38.698360	2	Restaurant	C	2	07:18.927029
2	Flights	C	2	08:42.389884	2	Clinic	M	6	07:45.403075
1	Flights	M	16	10:18.809487	2	Restaurant	C	10	08:56.765163
1	Deliveries	C	13	10:25.359836	1	Clinic	C	16	09:20.284593
2	Flights	C	9	10:51.761493	1	Restaurant	M	7	09:23.574403
1	Flights	M	7	10:52.183247	1	Restaurant	M	4	09:25.089084
2	Flights	C	10	10:52.495216	2	Restaurant	C	14	09:27.112225
1	Deliveries	C	8	11:39.151434	2	Restaurant	C	3	09:55.736324
1	Deliveries	C	15	12:03.519008	1	Clinic	C	15	10:25.475603
1	Flights	M	4	12:17.693128	2	Restaurant	C	5	10:37.460834
2	Flights	C	3	12:26.993837	2	Restaurant	C	9	10:49.014842
2	Flights	C	14	12:49.585392	1	Restaurant	M	16	10:56.743477
2	Flights	C	11	13:04.272941	1	Clinic	C	13	11:04.485390
1	Deliveries	C	4	13:16.470523	1	Clinic	C	4	12:06.690347
1	Deliveries	C	1	15:47.376327	1	Clinic	C	8	13:38.014602
1	Deliveries	C	16	18:02.259692	1	Clinic	C	12	14:37.197260
1	Flights	M	12	20:03.920754	1	Restaurant	M	1	17:09.073104
2	Flights	C	5	21:32.272442	2	Restaurant	C	11	17:11.980052
2	Flights	C	6	23:10.727760	1	Clinic	C	7	19:35.816561
1	Deliveries	C	7	23:20.991158	2	Restaurant	C	6	28:02.391335
1	Deliveries	C	12	41:29.414342	1	Clinic	C	1	28:18.301114

timings in seconds are visible. The bars for conventional technique and model tool use are paired for each participant, allowing easier visualization.

An important information found on the Primary Study is that there is not a single participant that could reuse the framework faster by using the conventional process than by using the reuse tool. The Secondary Study has provided similar results, only a single participant was able to be faster by using the conventional technique.

On Table 4 there are average timings and their proportions. By dividing the average time spent during the conventional process by the average time spent during model-based process, the result implies that the conventional technique took approximately 97.64 % longer than the model based tool.

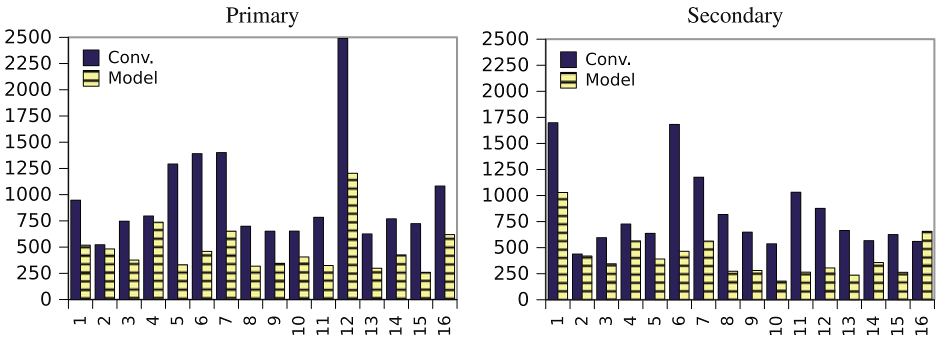


Fig. 4. Reuse Process Timings Bars Graph.

Table 4. Average Timings.

G.	Tech.	Avg.	Avg.(tech.)	Percents
1	Conventional	18:18.613745	32:25.698286	66.4026 %
2		14:07.084541		
1	Model based	09:46.65831	16:24.454048	33.5974 %
2		06:37.795738		
	Total	48:50.152334		100.0000 %

Table 5. T-Test Results.

T-Test	Data	Means	d.f.	t	p
Paired	Real	488.4596	15	5.841634	$3.243855{\cdot}10^{-05}$
Paired	Spare	417.8927	15	5.285366	$9.156136{\cdot}10^{-05}$
Two-Ssided	Both	771.4236 409.4295	43.70626	6.977408	$1.276575{\cdot}10^{-08}$

5.4 Hypothesis Testing

We applied Paired T-Tests for each of the presented studies and another T-Test after removing eight outliers. The seconds spent were processed using the statistic computation environment "R" [22]. The results of the T-Tests are shown on Table 5. The first column contains the type of T-Test, the second indicates the source of the data, the "Means" column indicate the resultant mean, which is the mean of the differences for an paired T-Test and one mean for each set for the other T-Test, which represent the conventional and the model based tool means, respectively. The "d.t." column stands for the degree of freedom; "t" and "p" are variables considered in the hypothesis testing.

The Paired T-Test is used to compare the differences between two samples related to each participant, in this case, the time difference of every participant is considered individually, and then, the means of the differences are calculated. The other T-Test is not paired, the means are calculated for the entire group, because a participant may be an outlier in a specific technique, which breaks the

Table 6. Chi-squared test for outlier detection.

Study	T.	G.	X^2	p	Position	Outlier
Real	C	1	5.104305	0.02386654	Highest	2489.414342
		2	2.930583	0.08691612	Highest	1390.72776
	M	1	4.091151	0.04310829	Highest	1203.920754
		2	2.228028	0.1355267	Highest	482.570996
Spare	C	1	4.552248	0.03287556	Highest	1698.301114
		2	5.013908	0.02514448	Highest	1682.391335
	M	1	3.917559	0.04778423	Highest	1029.073104
		2	2.943313	0.08623369	Lowest	179.467569

pairs. It is referred as two-sided because the two sets have the same number of elements, since the same number of outliers were removed from each group.

A "Chi-squared test" was applied in order to detect the outliers that were removed when calculating the last T-Test, which is referred as "Two-sided". The results of the "Chi-squared test" are found on Table 6. The 'M' in the techniques column indicates the use of our tool while 'C' indicates the conventional technique, the group column indicates the number of the group; the X^2 indicates the result of an comparison to the variance of the complete set and the position column indicates their position on the set, i.e., highest or lowest. The "outlier" column shows the timings in seconds that were considered abnormal.

In order to achieve better visualization of the outliers, we also provide line graphs. there are two line graphs in Fig. 5 which may be also used to visualize the dispersion of the timing records. In these plots, the timings for each technique are ordered independently, therefore, the participant numbers in these plots are not related to their identification codes.

According to the analysis from Table 5, since all p-values are less than the margin of error (0.01 %), which corresponds to the established significance level of 99.99 %, then, statistically, we can reject the "H0" hypothesis that states the techniques are equivalent. Since every t-value is positive, we can accept the

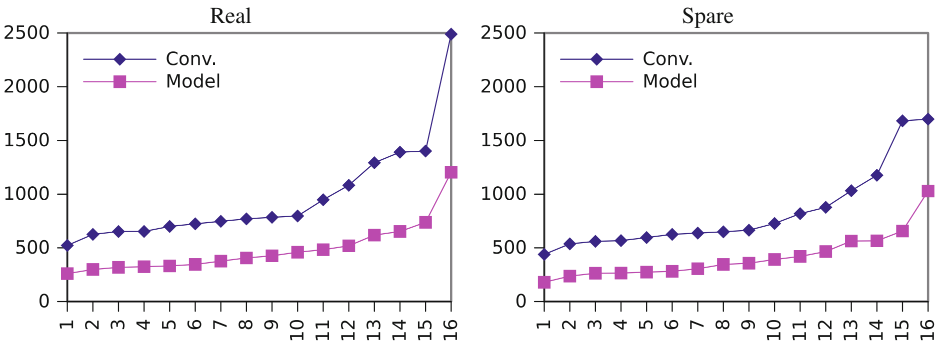


Fig. 5. Reuse Process Timings Bars Graph.

"Hp" hypothesis, which considers that the conventional technique takes more time than our tool.

5.5 Threats to Validity

The varied participant knowledge that could affect the collected data. To mitigate this threat, we divided the participants in two balanced groups considering the experience level and rebalanced the groups considering the preliminary results.

Students often tend to think they are being evaluated by experiment, which could affect the results. In order to mitigate this, we explained to the students that no one was being evaluated and their participation was considered anonymous.

Different computers and installations could affect the recorded timings. However, the different groups used equivalent computers in equal numbers and the participants were not allowed to change their machines during the same activity.

The participants already knew the researchers and knew that the model based tool was supposed to ease the reuse process. In order to avoid impartiality, we enforced that the participants had to keep a steady pace during the whole study.

It is possible that the reuse exercises are not accurate for every reuse of a crosscutting framework for real world applications. Only a single crosscutting framework was considered and the base applications had the same complexity. To mitigate this threat, the exercises were designed considering applications based on the real world.

To improve statistical reliability, we applied three T-Tests to statistically analyze the experiment data.

6 Conclusions

Considering the advantages of reducing the time needed to develop an information system, in this article, a model based process was presented, which raises abstraction levels of CF reuse. It serves as a graphical view that replaces textual cookbooks and is used to perform the reuse in a model driven approach. From our proposed model-based approach, a new reuse process was delineated, which allows engineers to start the reuse since earlier software development phases and reduce the time to reuse a CF.

Also, a new tool was developed to support the reuse process, which allows visualization of the form and is capable of transforming the models in order to generate the reuse code. With this, application developers do not need to worry about reuse coding issues nor how the framework was implemented, allowing to focus on the reuse requirements in a higher abstraction level.

We have conducted experiments that indicate that our tool has advantages on reducing the time to reuse a CF. With our tool, it is possible to develop information systems that reuse crosscutting frameworks in less time than by reusing the frameworks conventionally, which gives advantages to companies that rely on these systems.

It is also important to point that our tool is part of a project to develop an integrated development environment for frameworks, which currently supports CF feature subset selection and a CF repository service.

However, it was not yet analyzed if the tool brings advantages when maintaining an existing software nor if the tool may lead to more or less errors during development, which encourages us to conduct more experiments. We also need to evaluate how to deal with coupling multiple crosscutting frameworks to a single base application. Despite this functionality already being supported, some frameworks may conflict with each other and lead to unwanted results.

The code generated is based on AspectJ and it was not evaluated if it supports every CF without modifications. Although not stated, we have also worked on selecting subsets of features of the framework.

Acknowledgments. Thiago Gottardi would like to thank FAPESP (Process 2011/04064-8) and CNPq (Processes 132996/2010-3 and 560241/2010-0) for funding. Also, this paper was created inside a Universal Project granted by CNPq (Process Number 483106/2009-7).

References

1. Fayad, M., Schmidt, D.C.: Object-oriented application frameworks. Commun. ACM **40**, 32–38 (1997)
2. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., Irwin, J.: Aspect-oriented programming. In: Akşit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)
3. Camargo, V., Masiero, P.: Frameworks orientados a aspectos. In: Anais Do 19 Simpsio Brasileiro De Engenharia De Software (SBES'2005), Uberlndia-MG, Brasil, Outubro (2005)
4. Mortensen, M., Ghosh, S.: Creating pluggable and reusable non-functional aspects in Aspect C++. In: Proceedings of the Fifth AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (2006)
5. Shah, V., Hill, V.: An aspect-oriented security framework: lessons learned. In: Proceedings of AOSDSEC'04 (AOSD Technology for Application-Level Security). Workshop of the Aspect Oriented Software Development Conference, Lancaster, UK, 23 Mar 2004
6. Soares, S., Laureano, E., Borba, P.: Distribution and persistence as aspects. Softw. Pract. Exp. **33**(7), 711–759 (2006)
7. Kulesza, U., Alves, V., Garcia, A., de Lucena, C.J.P., Borba, P.: Improving extensibility of object-oriented frameworks with aspect-oriented programming. In: Morisio, M. (ed.) ICSR 2006. LNCS, vol. 4039, pp. 231–245. Springer, Heidelberg (2006)
8. Huang, M., Wang, C., Zhang, L.: Towards a reusable and generic aspect library. In: Workshop of the Aspect Oriented Software Development Conference at AOSDSEC'04, Lancaster, UK, 23 Mar 2004
9. Zanon, I., Camargo, V.V., Penteado, R.A.D.: Reestructuring an application framework with a persistence crosscutting framework. INFOCOMP **1**, 9–16 (2010)

10. Lazanha, R., Oliveira, A., Penteado, R., Ramos, R., Pastor, O., Camargo, V.: Uma arquitetura de referncia baseada em papis para frameworks transversais de persistncia: Uma anlise quantitativa. In: XXXVI Clei - Conferncia Latino-Americana de Informtica, Assuno, Paraguay (2010)
11. Bynens, M., Landuyt, D., Truyen, E., Joosen, W.: Towards reusable aspects: the mismatch problem. In: Workshop on Aspect, Components and Patterns for Infrastructure Software (ACP4IS'10), pp. 17–20 (2010)
12. Sakenou, D., Mehner, K., Herrmann, S., Sudhof, H.: Patterns for re-usable aspects in object teams. In: Net Object Days, Erfurt (2006)
13. Cunha, C., Sobral, J., Monteiro, M.: Reusable aspect-oriented implementations of concurrency patterns and mechanisms. In: Aspect-Oriented Software Development Conference (AOSD'06), Bonn, Germany (2006)
14. Soudarajan, N., Khatchadourian, R.: Specifying reusable aspects. In: Asian Workshop on Aspect-Oriented and Modular Software, Development (AOAsia'09) (2009)
15. France, R., Rumpe, B.: Model-driven development of complex software: a research roadmap. In: Future of Software Engineering, FOSE '07, Washington, DC, USA, pp. 37–54. IEEE Computer Society (2007)
16. Schmidt, D.C.: Model-driven engineering. IEEE Comput. **39**(2), 25–31 (2006)
17. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer, New York (2007)
18. Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, W.G.: An overview of aspectJ. In: Lindskov Knudsen, J. (ed.) ECOOP 2001. LNCS, vol. 2072, pp. 327–353. Springer, Heidelberg (2001)
19. Camargo, V.V., Masiero, P.C.: A pattern to design crosscutting frameworks. In: Proceedings of the ACM symposium on Applied computing, SAC '08, New York, NY, USA, pp. 759–764. ACM (2008)
20. Eclipse Consortium: Graphical Modeling Framework, version 1.5.0. http://www.eclipse.org/modeling/gmp/ (2011)
21. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell (2000)
22. Free Software Foundation, Inc.: http://www.r-project.org/, Dec 2011
23. Cechticky, V., Chevalley, P., Pasetti, A., Schaufelberger, W.: A generative approach to framework instantiation. In: Pfenning, F., Macko, M. (eds.) GPCE 2003. LNCS, vol. 2830, pp. 267–286. Springer, Heidelberg (2003)
24. Oliveira, T.C., Alencar, P., Cowan, D.: Reusetool-an extensible tool support for object-oriented framework reuse. J. Syst. Softw. **84**(12), 2234–2252 (2011)
25. Oliveira, T.C., Alencar, P.S.C., de Lucena, C.J.P., Cowan, D.D.: RDL: a language for framework instantiation representation. J. Syst. Softw. **80**, 1902–1929 (2007)

Large-Scale Enterprise Systems: Changes and Impacts

Wen Chen(✉), Asif Iqbal, Akbar Abdrakhmanov, Jay Parlar, Chris George, Mark Lawford, Tom Maibaum, and Alan Wassyng

McMaster Centre for Software Certification, McMaster University, Hamilton, Canada
chenw36@mcmaster.ca
http://www.mcscert.ca/

Abstract. Changes and their impacts to large-scale enterprise systems are critical and hard to identify and calculate. This work focuses on analysing changes and their potential impacts, and in particular on how regression testing following such changes can be minimised. The target scope of the approach we describe here is systems containing hundreds of thousands of classes and millions of methods. It is extremely difficult and costly to apply regular regression testing techniques to such systems. It is very expensive and often unnecessary to retest everything after a change is introduced. Selective retesting is dangerous if the impacts of change are not understood, and analysing such systems to understand what is being changed and what the impacts are is difficult. This paper proposes a way to perform a change impact analysis which makes it possible to do efficient, targeted regression testing of enterprise systems. Our approach has been tried on a large system comprising 4.6 million methods with 10 million dependencies between them. Using our approach, maintainers can focus on a smaller, relevant subset of their test suites instead of doing testing blindly. We include a case study that illustrates the savings that can be attained.

Keywords: Large-scale enterprise systems · Impact analysis · Static analysis · Dependency graph

1 Introduction

Enterprise systems are typically large, complicated, and may also be inadequately documented and date back a number of decades. As a consequence they are also often legacy systems: poorly understood and difficult to maintain. To make matters worse, they are often mission critical, being found in critical roles as strategic systems in large companies. So they are typically seen as both critical and fragile.

Patches are supplied by vendors to the underlying middleware, and for a number of reasons may need to be applied. The latest IT Key Metrics Data from Gartner [1] report that in 2011 some 16 % of application support activity

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 274–290, 2013.
DOI: 10.1007/978-3-642-40654-6_17,

was devoted to technical upgrades, rising to 24 % in the banking and financial services sector. Hardware and operating systems change. The user organization develops new or changed requirements that need to be implemented. A perpetual problem for the organization is how to manage such changes with minimum risk and cost.

Risk of unintended change is typically addressed by regression testing. The problem is that regression testing can be expensive and time-consuming for large systems with interactive interfaces. Organizations can spend millions of dollars per annum on it. The actual effect of a middleware patch or an application software change may in fact be minimal, so a small fraction of the regression tests may be sufficient; but, with an enterprise system, it is very risky to make a judgement about what should be tested and what can be assumed to be OK.

What is needed is a way to identify the *impact* of any change. What business processes might be affected by a patch to the middleware, or by a planned change to the application software, or the way data is stored in the database? If organizations know the possible impact of a change they can select only the relevant regression tests, confident that the others do not need to be run, because the results will not change.

The techniques and tools we have developed are based on static analysis of the code in the system and the code in the patch. Static analysis may be compared with the informal approach of reading the documentation, or with the formal one of dynamic analysis, where the code is instrumented in some way to generate output, logs in particular, that show what it is doing. Documentation, especially for legacy systems, may be incomplete, misleading, or just plain wrong. Dynamic analysis is precise, but essentially incomplete in the same way that testing is, and for the same reason: unless you have run your system with all possible inputs, you cannot know if you have found all possible behaviours. Static analysis looks for possible behaviours, or (in our case) possible dependencies. It typically finds too many, depending on how fine grained the analysis is, and how sophisticated it is, but what it rules out can be ruled out for certain. For example, if method A mentions in its code methods B and C, then we decide that A depends on B and C. It may be that, in practice, A never calls B, only C, but unless our tools can be certain about this, they take the conservative approach of assuming that if either B or C change, A may change. What we can be certain about is that A cannot change just because some other method D changes (unless the recursive analysis of B or C leads us to it). Static analysis is accurate, unlike documentation, because it is the actual system code we analyze, not a description of it. It is complete, unlike dynamic analysis, because it takes the conservative but safe approach outlined.

The approach is conventional. First, we calculate the dependencies between methods and fields in the existing middleware library and user's application, between objects in the database, and between database objects and program methods: *dependency analysis*. Second, we identify what is changed by a patch: *patch analysis*. Third, we calculate, by starting with the things changed in the patch and following the dependencies in reverse, what might be changed when the patch is applied: *impact analysis*.

The things that might be changed, the *affected* methods or methods, can stretch right through the middleware library and the user's application software. In practice, we will only be interested in some of these, usually those in the application software that appear in test cases. Identifying these means the user can identify the regression tests that will need to be applied and hence, as test cases are usually grouped by business process, which business processes may be affected. The methods we select we term the *methods of interest*: the precise way we identify these will depend on how the test cases are organized.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes dependency analysis. Patch analysis is introduced in Sect. 4, and impact analysis in Sect. 5. Section 6 describes a case study. Section 7 summarizes the paper, draws some conclusions, and outlines future work.

2 Related Work

Change Impact Analysis applied to software systems can be traced back to the 1970s. Reasons for doing change impact analysis are well known and understood: "As software components and middleware occupy more and more of the software engineering landscape, interoperability relationships point to increasingly relevant software change impacts." [2] Moreover, due to the increasing use of techniques such as inheritance and dynamic dispatching/binding, which come from widely used object-oriented languages, small changes can have major and nonlocal effects. To make matters worse, those major and nonlocal effects might not be easily identified, especially when the size of the software puts it beyond any maintainer's ability to adequately comprehend.

There is considerable research related to this field, but it seems that there are limited known ways of performing change impact analysis. Bohner and Arnold [2] identify two classes of impact analysis: traceability and dependency. What we are interested in this work is dependency: linkages between parts, variables, methods, modules etc. are assessed to determine the consequences of a change.

Dependency impact analysis can be either static, dynamic or a hybrid of the two. We discuss some of the work using these techniques below.

Static impact analysis [2–6] identifies the impact set - the subset of elements in the program that may be affected by the changes made to the system. For instance, *Chianti* [3] is a static change impact analysis tool for Java that is implemented in the context of the *Eclipse* environment, which analyzes two versions of an application and decomposes their differences into a set of atomic changes. The change impact is then reported in terms of affected tests. This is similar to our approach, but lacks the capability to deal with the database components.

Apiwattanapong et al. [7] argue that static impact analysis algorithms often come up with too large impact sets due to their over conservative assumptions: the actual dependencies may turn out to be considerably smaller than the possible ones. Therefore, recently, researchers have investigated and defined impact analysis techniques that rely on dynamic, rather than static, information about program behaviour [8–11].

The dynamic information consists of execution data for a specific set of program executions, such as executions in the field, executions based on an operational profile, or executions of test suites. [7] defines the dynamic impact set to be the subset of program entities that are affected by the changes during at least one of the considered program executions. *CoverageImpact* [8] and *PathImpact* [12] are two well known dynamic impact analysis techniques that use dynamic impact sets. *PathImpact* works at the method level and uses compressed execution traces to compute impact sets. *CoverageImpact* also works at the method level but it uses coverage, rather than trace, information to compute impact sets. Though the dynamic approach can make the analysis more efficient, it doesn't guarantee that all system behaviors can be captured by it. Thus it might cause a good number of *false negatives*, i.e. potential impacts that are missed.

Recently, a hybrid of static and dynamic analysis is being investigated. [13] proposes a hybrid technique for object-oriented software change impact analysis. The technique consists of three steps: static analysis to identify structural dependencies between code entities, dynamic analysis to identify dependencies based on a succession relation derived from execution traces, and a ranking of results from both analyses that takes into account the relevance of dynamic dependencies. The evaluation of this work showed it produced fewer false negatives but more false positives than a precise and efficient dynamic tool *CollectEA* [7].

The *Program Dependency Graph* (PDG) [14] and associated *Slicing* [15] techniques work at the statement level. We need to work at the level of methods and fields because of the size of the program being analyzed. Rothermel and Harrold [16] identified two kinds of *Regression Test Selection* (RTA) techniques: *minimization* and *safe coverage*. *Minimization* selects minimal sets of tests through modified or affected program components, while *safe coverage* selects every test in the test suite that may test the changed parts of the program. According to the definition, our work is a *safe coverage* approach. In our problem domain, precision is less important than safety.

The size of the program we want to analyse is a major factor driving our approach. The sizes of the software systems most current impact analysis associated techniques [8,17,18] are dealing with are orders of magnitude smaller than the enterprise systems we have targeted. Taking *DEJAVOO* [19] as an example, the largest system analyzed in the empirical study was JBOSS, which contains 2400 classes. As the two-phase process takes a considerable time to complete, systems of the size we are concerned with were clearly beyond the scope of that research. We have not found any related work that claims to handle such large systems.

Our approach is based on static analysis (a) because of the feasibility issue for large programs and (b) because we consider false negatives (missed impacts) much more dangerous than false positives (identified impacts which in practice cannot occur). The very large size of the middleware component means that in practice only small parts of it are likely to be exercised by a particular user, and so even a coarse analysis can produce dramatic savings in regression testing. In addition, dynamic analysis requires run-time information from running test suites and/or actual executions, which may not be available, may be very expensive to produce, and may interfere with the execution and so produce spurious results.

3 Dependency Analysis

We consider a system having, typically, three layers: (i) the user's application code, (ii) the middleware library, and (iii) a database. Our intention is to construct a *dependency graph* which we can use to calculate the potential impact of a change within such a system. In general, if item A refers to item B, then a change to B has a potential effect on A: A is *dependent* on B. A and B might be database tables, other database objects such as stored procedures or triggers, or methods or fields in the application code or library. In practice we divide this analysis into three parts, which we will discuss in turn. *Program dependencies* are between methods and/or class fields within the application code and library (which we analyse as a single program). *Program-database dependencies* are between the program and the database. *Database dependencies* are between database objects.

3.1 Program Dependencies

The system for which we have so far developed tools is written in Java, so we will use Java terminology and discuss the particular problems an object-oriented language supporting dynamic binding introduces.

Method invocations are candidates for dynamic binding, meaning that compile time calling of a method might cause calling of another method at runtime, due to class inheritance, interface implementation and method overriding. We illustrate the dynamic problem with two examples. Consider the situation in Fig. 1 where classes B and C override class A's method $m()$. Although statically all three calls are to $A.m()$, dynamically they redirect to $A.m()$, $B.m()$ and $C.m()$, respectively.

As a second example, consider Fig. 2 where classes B and C don't override class A's $m()$ method. Here, the compile time call to $C.m()$ redirects to $A.m()$ at runtime.

One way to handle dynamic binding statically is to include all classes from the inheritance hierarchy, as in *Class Hierarchy Analysis* (CHA) [20]. The drawback of this approach is the huge number of redundant call edges that might result: it creates an edge from each caller of a method m to every possible instance of m. Consider Fig. 3 where the edges to $C.foo()$ are redundant because only $A.foo()$ and $B.foo()$ have real bodies defined.

Similarly, in Fig. 4 the edge to $B.foo()$ is redundant because A is actually the closest transitive superclass of C that has a body of method $foo()$ defined.

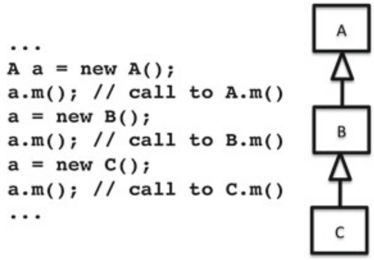


Fig. 1. Dynamic Binding Example 1.

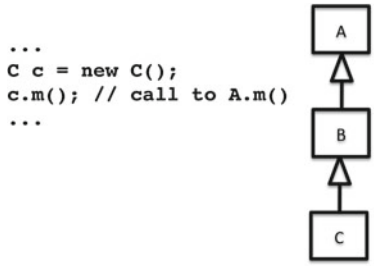


Fig. 2. Dynamic Binding Example 2.

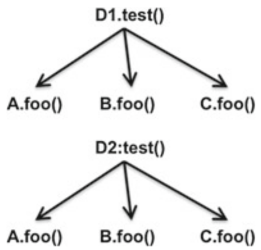
In the java library we analysed there was an interface with more than 50,000 transitive subclasses. If there were, say, 100 callers of a method of this interface, 5 million edges would be generated. In practice we found that only a few dozen of the transitive subclasses would define a particular method, and a more precise analysis could save perhaps 99 % of these edges.

Some techniques like *Rapid Type Analysis* (RTA) and *Variable Type Analysis* [20] do exist to tackle this problem and we tried these approaches using the tool Soot [21], but had to abandon it due to excessive memory consumption and memory overflow problems. These approaches turned out be unsuitable for our huge domain. This is one of the reasons we resorted to a new technique which we call *access dependency analysis*.

The full details of access dependency analysis are in a technical report [22], but we illustrate it with our two examples. Consider Fig. 5. The graph shown is the dependency graph resulting from the access dependency analysis of the code shown in Fig. 3a. Note that since $C.foo()$ has no real body of its own, it is not in the graph. We only consider the overridden versions of methods during the addition of extra edges for handling dynamic binding, which reduces the number of edges. Also instead of adding call edges from $D1.test()$ and $D2.test()$ to $B.foo()$, we add an edge from $A.foo()$ to $B.foo()$. What this edge implies is that a compile time call to $A.foo()$ *might* result in a runtime call to $B.foo()$. This kind of edge reduces the number of edges even further because each additional caller only increases the number by one (like the edge from $D2.test()$ to $A.foo()$).

```
Class A{
        public void foo(){
        ...
        }
}
Class B extends A{
        public void foo(){
        ...
        }
}
Class C extends B{
        //does not override foo()
}
Class D1{
        public void test(){
                A a = new A();
                a.foo();
        }
}
Class D2{
        public void test(){
                A a = new B();
                a.foo();
        }
}
```

(a) Sample code segment



(b) Graph generated

Fig. 3. Conservative Analysis Example 1.

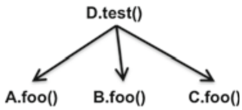
As for the second example, consider Fig. 6 where the graph shown is the dependency graph resulting from the access dependency analysis of the code shown in Fig. 4a. Here, since A is the closest transitive superclass of C for the function $foo()$, a compile time call to $C.foo()$ redirects to $A.foo()$, and we don't include $B.foo()$ in the graph. The result is, once again, a reduced number of edges.

We see that we get an efficient dependency graph because (i) links for each overridden method are only included for the actual overrides and (ii) the size of the graph grows linearly with the number of callers.

As mentioned earlier, a class with over 50,000 transitive subclasses was found to have only a few dozen of them which override a particular method. Using our access dependency analysis, we only get a few hundred edges (rather than almost 5 million edges generated by the conservative analysis). Since the number

```
Class A{
        public void foo(){
        ...
        }
}
Class B extends A{
        //does not override foo()
}
Class C extends B{
        //does not override foo()
}
Class D{
        public void test(){
                C c = new C();
                c.foo();
        }
}
```

(a) Sample code segment



(b) Graph generated

Fig. 4. Conservative Analysis Example 2.

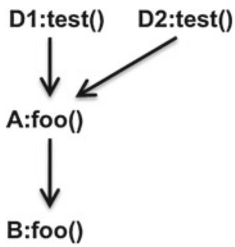


Fig. 5. Access Dependency Analysis Example 1.

of edges are reduced, we also get rid of the memory overflow problem we faced in applying other existing approaches.

3.2 Program-Database Dependencies

Program code interacts with the database via SQL commands generated as strings, and passed to the database through the SQL-API. Such strings are often dynamically created, so we used a string analysis tool, the Java String Analyzer [23] (JSA), which is capable of statically analyzing a section of code and determining *all* the strings which might possibly occur at a given string expression, including dynamically constructed strings. Currently the JSA can only work with Java code, but it is architected in such a way that a single layer

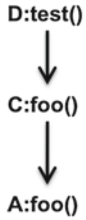


Fig. 6. Access Dependency Analysis Example 2.

of it can be replaced to add support for a different language, leaving the majority of the JSA unchanged.

Ideally, all of the Java classes in a program would be passed simultaneously to the JSA, all usage of the SQL would be checked, and a report would be provided showing which strings, and therefore which database object names, are possible for each SQL string. Unfortunately, the process used by the JSA is extremely resource intensive. Using just a small number of classes (∼50) will often cause memory usage to explode. On a 32 GB RAM machine, all available memory was quickly exhausted in many tests.

A way to segment the classes into small sets which the JSA can handle is thus required. The technique used for this was to identify all of the unique "call-chains" which exist in the program, for which the bottom-level of a chain is any method which makes use of SQL, and the top-level of the chain is any one of the "methods of interest". These chains can be constructed via analysis of the program dependency graph.

Using this technique on a large Java program, "call-chain explosion" was quickly encountered, due to cycles in the dependency graph. Initially we were using a modified depth-first search to go through the graph. Traditional depth-first search algorithms are only concerned with finding if one node can be reached from another. Generating a listing for each path is not their goal. We needed a modified version because *all* possible call-chains through the graph were required. This modified version ended up falling victim to the cycles, resulting in an infinite number of chains.

To solve this, Tarjan's Algorithm [24] was employed to identify all the strongly connected components (i.e. cycles). These components were then compacted into a single node, preserving all of the incoming and outgoing edges of all nodes in the strongly connected component. The modified depth-first search is then run, recording all the possible SQL-related call-chains through the graph.

This approach still has two problems. First, some classes cause the JSA to fail. We built up a list of such classes and adapted JSA to exclude them from the analysis. Second, some strings are effectively just wild cards, because they depend on user inputs, or because of incomplete analysis (the use of call chains instead of complete call trees, and the exclusion of some classes). Again we take the conservative approach, and assume that a wildcard string can be dependent on any database object. Fortunately, the proportion of these in practice seems to be quite small.

3.3 Database Dependencies

The third component of our dependency graph are dependencies between database objects: tables, triggers, procedures, etc. Fortunately, we had a database in which such dependencies were already stored. Otherwise it would be fairly straightforward to write some SQL procedures to calculate them.

4 Patch Analysis

As software evolves, there are incremental changes to an existing, perhaps large, set of code and documentation [25]. Users often *have* to apply vendor patches to potentially fix issues or ensure continuing vendor support.

A single patch can consist of multiple files, and depending on its type, a patch may update the library or database (or both) of a system. Patches typically contain a large number of different types of files, such as program code, SQL, and documentation. It is necessary to distinguish between files that will change the program or database, and files that will not, and for the first category be able to parse them to see which methods, tables, procedures etc. may be changed. Vendor documentation of the patch is typically inadequately detailed for this task. It is also better to rely on the source files themselves than on the accuracy and completeness of the documentation.

4.1 Database Changes

We need to identify which changes are capable of affecting database objects. Major changes to Oracle databases (for example), come from SQL and PL/SQL scripts. To do this, we employed an SQL parser to capture the names of those objects. We had to extend the original tool to deal with SQL statements that themselves contained SQL definitions.

Other patch files also use SQL for making their changes. In some the SQL is contained in other text, in others it is compiled and has to be decompiled to extract the SQL for the parser. We developed a suite of tools to handle all the relevant files found in Oracle patches.

4.2 Library Changes

Patches to Java libraries often come in the form of class and jar files, and techniques are necessary for detecting changes at the method and field levels between the original software and the patch.

Some tools and techniques do exist to detect difference between two versions of a program. The Aristotle research group from Georgia Tech. [17] showed an approach for comparing object-oriented programs. Their approach was not

```
class Test{
    int i;
    public void foo(){
       i++;
    }
    public void bar(){
       System.out.println();
    }
}
```

(a) Original Code

```
class Test{
    int i;
    public void foo(){
       i++;
       bar();
    }
    public void bar(){
       System.out.println();
    }
}
```

(b) Modified Code

```
<changed signature="foo()">
    <methodinfo>
       <instructions/>
      <dependencies>
          <addedcall>Test:bar()</addedcall>
      </dependencies>
    </methodinfo>
</changed>
```

(c) XML Segment Describing Difference

Fig. 7. Detecting Modifications

applicable to our domain because it compares both versions of the whole program, rather than making individual class to class comparisons. Moreover, their application domain was several orders of magnitude smaller than ours. Meanwhile, there exist some open source tools like *JDiff* [26], *Jar Comparer Tool* [27], *JarJarDiff* and *ClassClassDiff* [28] which only give API differences between two *class* or two *jar* files. To achieve the level of detail we require, namely which methods and fields are changed, we decided to write our own modification detecting tool for class and jar files. This operates by first converting the Java to XML. We compare the two versions of an XML file, node by node, to detect differences in methods, fields, access flags, superclass, interfaces, etc. and list them in another XML file for use in the impact analysis phase.

As an example, Fig. 7a, b show an original and modified code sample, respectively. Note that the only difference is the inclusion of the call to method $bar()$ inside method $foo()$. The XML segment in Fig. 7c contains this modification information.

5 Impact Analysis

Our aim is to identify a subset of the previously selected *methods of interest*, typically methods appearing in test suites, that are affected by a directly changed method, field, or database object that has been identified by patch analysis (or, perhaps, is a candidate for change by the user).

The overall process, illustrated in Fig. 8, is:

1. We create the program dependency graph. The graph edges are in fact the reverse of the dependency relation, from each method or field to those methods which depend on it, because that is the direction in which we search.

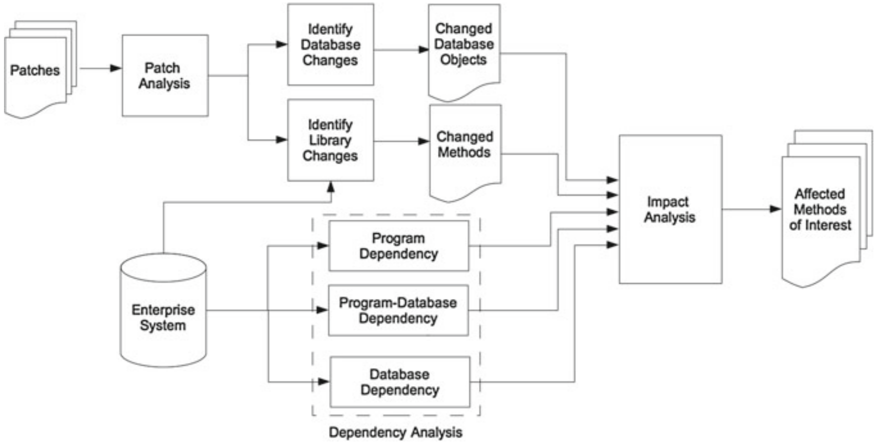


Fig. 8. System Flow Chart.

2. We similarly build the database dependency graph.
3. For each instance of the SQL-API, we use the program dependency graph to identify call chains which terminate in a method of interest, and so create a relation between methods of interest and SQL strings. This is the program-database dependency.
4. We use patch analysis to identify changed methods and fields in the library, and changed database objects. (If impact analysis is being used to investigate the impact of proposed changes by the user to application code and/or database objects, patch analysis is replaced by analysis of the design of the proposed changes to see what existing methods, fields, and database objects are to be changed.)
5 To calculate the dependencies on a database object, we proceed as follows:
   (a) We calculate the reflexive transitive closure of the dependents of our element using the database dependency graph, a set S, say.
   (b) For each element in S we find each SQL string in the program-database dependency that can include its name as a substring, and for each such add the corresponding method of interest to our results.
6. To calculate the dependencies on a program method or field, we simply search the program dependency graph to find all dependents of the method or field, noting any methods of interest that we encounter.
7. The last two steps constitute *impact analysis*. In either case, the result is a subset of the methods of interest.

Impact analysis is illustrated in Fig. 9. In this figure, stars are methods of interest, circles are library methods and fields, and squares are database objects. Directly changed items (methods, fields, or database objects) are black, and potentially affected ones are grey. Unaffected items are white. The dashed lines represent dependencies between items. Note the dependencies may occur not only between items of different layers, but also between items within the same

Fig. 9. Impact Analysis.

layer. For simplicity and clarity, we only show dependencies between layers here. We are searching for the grey stars, the potentially affected methods of interest.

6 Case Study

For a case study we took a particular (recent) vendor patch for a commonly used middleware system. The library is written in Java, and contains some 230,000 classes, and over 4.6 million methods. The database as supplied by the vendor contains over 100,000 objects: tables, triggers, procedures, etc.

The calculation of the program dependency graph took over 7 h on a quad core 3.2 GHz machine with 32 GB RAM running 64-bit Linux. This is a little above the average desktop, but by no means a supercomputer. The time is large, but quite manageable, especially as this analysis is independent of any patch or proposed change, and can be prepared in advance. This graph forms a substantial corporate asset for other kinds of analysis, and can be easily and quickly updated as the system changes, provided we do the proper analysis of the changes. The dependency graph has over 10 million edges. Searching this dependency graph takes only a few seconds for each starting point method or field.

The patch contained 1,326 files with 35 different file types. Among those 35 file types, 11 can possibly affect either the library or the database, or both. We ran our tools on each file with one of these 11 types and identified 1,040 directly changed database objects (to which database dependency analysis added no more) and just 3 directly changed Java methods.

The program-database dependency approach described in Sect. 3.2 found that 19,224 out of the 4.6 million methods had SQL-API calls, and that 2,939 of these methods (just over 15 %) had a possible dependency on one of the 1,040 affected database objects.

We adopted as our definition of "methods of interest" those which were not themselves called by anything else, "top callers", and there were 33,896 of these, a fraction over 2 % of all top callers. The patch, as might be expected, only affects a tiny part of the library, and we can identify that part, and do so in a short space of time.

7 Conclusions

7.1 Achievement

The achievement of this work is threefold:

- We have developed an improved dependency model for dealing with object-oriented languages like Java that support inheritance and dynamic binding, and shown it to be equivalent (in terms of finding static dependencies) to other techniques that typically create much larger dependency graphs.
- We have demonstrated the practical applicability of the improved model to a very large enterprise system involving hundreds of thousands of classes. Such systems may be perhaps 2 orders of magnitude larger than the systems analyzed by other approaches, so our technique seems to be uniquely powerful.
- We have developed the techniques of string analysis beyond those of the Java String Analyser we started with in order again to deal with large size, and to overcome its inability to deal with some of the classes we encountered.

The last point is typical of the work we have done, in developing existing tools to deal with large size, and in developing our own tools, techniques, and data structures to deal with the magnitude of the problem. This has been above all else an exercise in software engineering.

Change impact analysis is performed in three stages: *dependency analysis* with granularity at the level of method or field for program code and database object (table, trigger, procedure, etc) for the database component. The granularity choice is coarser than examining code at the statement level, or database tables at the attribute or data item level, but enables the technique to be used on large, real systems. Dependency analysis generates dependency relations between methods, fields, and database objects that can be searched. The second stage is *patch analysis*, the identification of changed methods, fields, and database objects. Third, *impact analysis* combines the first two outputs to identify the potentially affected *methods of interest* in the user application. If the *methods of interest* are chosen to be those methods appearing in test cases, then we can identify a subset of the regression tests that need to be rerun after a change. Current indications from our case study are that these subsets may be comparatively small, giving a consequent substantial reduction in the considerable costs, resources and time involved.

The analysis is conservative: while we would like to reduce false positives, we are determined that there will be no false negatives, i.e. potential impacts that remain undetected. We can not tell our users that running the reduced set of tests our analysis generates will find any problems caused by the changes: that will depend on the quality of their test sets. But we can tell them that running any more of their test sets will be a waste of resources.

7.2 Future Work

Organizations tend to identify their test suites by the business process that is being tested, and to think of their system as consisting of (or supporting) business processes rather than code classes. HP Quality Centre, for example, organizes tests by business process. By analysis of test cases we will be able to relate the affected methods of interest to the business processes that might be affected, and hence present results in a way that is more meaningful to testing departments.

In the medium term, there are a number of other related applications that can be achieved with the techniques we have developed. First, we need to extend the work beyond the current Java tools, to systems written in other languages such as COBOL. The modular design of our system, especially an analysis based on XML, means that only language-dependent front ends would be needed for each such extension.

We started out intending to analyse vendor-supplied patches. But we could have started out with any method, field or database object that the user might intend to change. We can then identify which existing tests might execute or depend on that selected item. This can help users improve test cases. Such work might be a prelude, and complementary, to dynamic analysis to examine test coverage. Indeed, our analysis makes such dynamic analysis feasible. The dependency graph identifies the possible methods that might be called from a given method. If you are testing that method, and want to have some idea of the coverage of your tests, the relevant baseline is the subgraph of the dependency graph with the method being tested at its apex, not the whole of the library, which is otherwise all you have. Any particular organization probably only uses a tiny fraction of the whole library, and the subgraph of the dependency graph containing that organization's methods of interest is the only part they need to be concerned with.

Finally, impact analysis can be used in planning enhancements to applications. Once methods or database objects that are intended to be changed are identified, typically in the detailed design stage, the same impact analysis as we use on changes caused by patches can be done to indicate where the potential effects are. This raises a number of possibilities. The testing necessary to cover all possible impacts can be planned. Or, perhaps, the design may be revisited to try to reduce the possible impact.

Acknowledgements. The authors are grateful to Ron Mison, Sundaram Viswanathan, and Vinayak Viswanathan of Legacy Systems International, for introducing us to the problem, and also for working with us to ensure that our research produces practical methods and tools. We also thank John Hatcliff for his advice and for pointing us to relevant work, and Wolfram Kahl for his technical advice throughout the project.

References

1. IT Key Metrics Data 2012. Gartner, Inc., December 2011
2. Bohner, S.A.: Software change impact analysis. In: Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering, Workshop (SEW-27'02) (1996)
3. Ren, X., Shah, F., Tip, F., Ryder, B.G., Chesley, O.: Chianti: a tool for change impact analysis of java programs. SIGPLAN Not. **39**, 432–448 (2004)
4. Pfleeger, S., Atlee, J.: Software Engineering: Theory and Practice. Prentice Hall, Englewood Cliffs (2006)
5. Ayewah, N., Hovemeyer, D., Morgenthaler, J., Penix, J., Pugh, W.: Using static analysis to find bugs. IEEE Softw. **25**, 22–29 (2008)
6. Khare, S., Saraswat, S., Kumar, S.: Static program analysis of large embedded code base: an experience. In: Proceedings of the 4th India Software Engineering Conference, ISEC '11, (New York, NY, USA), pp. 99–102. ACM (2011)
7. Apiwattanapong, T.: Efficient and precise dynamic impact analysis using execute-after sequences. In: Proceedings of the 27th International Conference on Software Engineering (2005)
8. Orso, A., Apiwattanapong, T., Harrold, M.J.: Leveraging field data for impact analysis and regression testing. In: Proceedings of the 9th European Software Engineering Conference held jointly with 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering, vol. 28(5), September 2003
9. Breech, B., Danalis, A., Shindo, S., Pollock, L.: Online impact analysis via dynamic compilation technology. In: 20th IEEE International Conference on Software Maintenance (2004)
10. Patel, C., Hamou-Lhadj, A., Rilling, J.: Software clustering using dynamic analysis and static dependencies. In: 13th European Conference on Software Maintenance and Reengineering, 2009. CSMR '09, pp. 27–36. March 2009
11. Li, H.: Dynamic analysis of object-oriented software complexity. In: 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 1791–1794, April 2012
12. Law J., Rothermel, G.: Incremental dynamic impact analysis for evolving software systems., In: Proceedings of the 14th International Symposium on Software, Reliability Engineering (2003)
13. Maia, M.C.O., Bittencourt, R.A., de Figueiredo, J.C.A., Guerrero, D.D.S.: The hybrid technique for object-oriented software change impact analysis. In: European Conference on Software Maintenance and Reengineering, pp. 252–255 (2010)
14. Ferrante, J., Ottenstein, K.J., Warren, J.D.: The program dependence graph and its use in optimization. ACM Trans. Program. Lang. Syst. **9**, 319–349 (1987)
15. Ottenstein, K.J., Ottenstein, L.M.: The program dependence graph in a software development environment. SIGPLAN Not. **19**, 177–184 (1984)
16. Rothermel, G., Harrold, M.: Analyzing regression test selection techniques. IEEE Trans. Softw. Eng. **22**, 529–551 (1996)
17. Apiwattanapong, T., Orso, A., Harrold, M.: A differencing algorithm for object-oriented programs. In: Proceedings of the 19th International Conference on Automated Software Engineering 2004, pp. 2–13. September 2004

18. Canfora, G., Cerulo, L.: Impact analysis by mining Software and Change Request Repositories. In: IEEE International Symposium on Software Metrics, pp. 20–29 (2005)
19. Orso, A., Shi, N., Harrold, M.J.: Scaling regression testing to large software systems. SIGSOFT Softw. Eng. Notes **29**, 241–251 (2004)
20. Bacon, D., Sweeney, P.: Fast static analysis of C++ virtual function calls. In: Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, ACM SIGPLAN Notices, vol. 31, pp. 324–341. ACM Press, New York, October 1996
21. Lam, P., Bodden, E., Lhotak, O., Lhotak, J., Qian, F., Hendren, L.: Soot: A Java Optimization Framework. Sable Research Group, McGill University, Montreal, Canada, March 2010. http://www.sable.mcgill.ca/soot/
22. Chen, W., Iqbal, A., Abdrakhmanov, A., George, C., Lawford, M., Maibaum, T., Wassyng, A.: Report 7: Middleware Change Impact Analysis for Large-scale Enterprise Systems. Tech. Rep. 7, McMaster Centre for Software Certification (McSCert), September 2011
23. Christensen, A.S., Møller, A., Schwartzbach, M.I.: Precise analysis of string expressions. In: Cousot, R. (ed.) SAS 2003. LNCS, vol. 2694, pp. 1–18. Springer, Heidelberg (2003). http://www.brics.dk/JSA/
24. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM J. Comput. **1**(2), 146–160 (1972)
25. Mockus, A., Weiss, D.M.: Predicting risk of software changes. Bell Labs Tech. J. **5**(2), 169–180 (2000)
26. Doar, M.B.: JDiff - An HTML Report of API Differences (2007). http://javadiff.sourceforge.net/
27. Jar Compare Tool. http://www.extradata.com/products/jarc/
28. Tessier, J.: The Dependency Finder User Manual, November 2010. http://depfind.sourceforge.net/Manual.html

Part V

Software Agents and Internet Computing

Bringing Diversity to Recommendation Lists – An Analysis of the Placement of Diverse Items

Mouzhi Ge[1(✉)], Dietmar Jannach[2], and Fatih Gedikli[2]

[1]Bundeswehr University Munich, Werner-Heisenberg-Weg 39, Neubiberg, Germany
mouzhi.ge@unibw.de
[2]TU Dortmund, Joseph-von-Fraunhofer-Straße 23, Dortmund, Germany
{dietmar.jannach, fatih.gedikli}@tu-dortmund.de

Abstract. The core task of a recommender system is to provide users with a ranked list of recommended items. In many cases, the ranking is based one a recommendation score representing the estimated degree to which the users will like them. Up to now research specifically focused on the *accuracy* of recommender algorithms in *predicting* the relevance of items for a given user. However, researchers agree that there are other factors than prediction accuracy which can have a significant effect on the overall quality of a recommender system. Therefore, additional and complementary metrics, including diversity, novelty, transparency and serendipity should be used to evaluate the quality of recommender systems.

In this paper we will focus on *diversity* which has been more widely discussed in recent research and is often considered to be a factor which is equally important as accuracy. In particular we address the question of how to place diverse items in a recommendation list and measure the user-perceived level of diversity. Differently placing the diverse items can affect perceived diversity and the level of serendipity. Furthermore, the results of our analysis show that including diverse items in a recommendation list can both increase and sometimes even decrease the perceived diversity and that the effect depends on how the diverse items are arranged.

Keywords: Recommender system · Evaluation · Diversity · Serendipity · Item ranking · User satisfaction

1 Introduction

Recommender systems are developed to help users find relevant products that may interest them. The goal of recommender systems is to reduce the information overload and provide personalized recommendations for users. Over the last decade, recommender systems have been widely applied in e-commerce, for example, book recommendation on Amazon or movie recommendation on Netflix [12]. Moreover, some case studies have stated that the use of recommender systems can both increase user satisfaction and produce added value to the business [6, 11, 21].

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 293–305, 2013.
DOI: 10.1007/978-3-642-40654-6_18,

As there is a growing popularity of using recommender systems in e-commerce, a variety of recommender algorithms have been proposed over the last fifteen years. Most of these algorithms focus on improving recommendation accuracy. Accordingly, the performance of recommender systems was evaluated by accuracy metrics such as Mean Absolute Error (MAE) or Precision and Recall. A recent literature survey shows that still today both the Information Systems and Computer Science community very strongly rely on these measures [13]. However, some researchers have proposed that being accurate alone is not enough [15]. Additional and complementary metrics, including diversity, novelty and serendipity as well as transparency could be used to evaluate the quality of recommender systems [4, 9, 10]. Among the proposed metrics, diversity has been widely discussed and considered to be a factor that is equally important as accuracy [7, 16].

The concept of diversity in recommender system research can be generally divided into *inherent* diversity and *perceived* diversity. Inherent diversity considers diversity from an objective view and is often measured by the dissimilarity among the recommended items [19, 20]. The set of recommended items can either refer to a single list of recommendations for a single user or the set of overall recommendations from the whole system. Thus the concept of inherent diversity comprises intra-list diversity as defined by [20] and aggregate diversity as proposed by [1]. While intra-list diversity means the diversity inside certain recommendation list, aggregate diversity refers to the diversity among the recommendations across all the users.

Perceived diversity, in contrast, defines diversity from a subjective perspective and can only be determined through a user evaluation. The advantage of focusing on perceived diversity is that we can directly capture the users' opinions. Lathia et al. [14] found that perceived diversity is positively related to user satisfaction in the long term when using a recommender system. Regarding the importance of perceived diversity, this paper will analyze how end users perceive the diversity-increasing items in recommendation lists. Our experimental study will use movie recommendations as an example. The diversity will be varied by adding movies from different genres.

One factor that may affect the perceived diversity but has not been analyzed in research so far, is the placement of diversity-increasing items in the recommendation list. Suppose we have several diverse items that we can include in a recommendation list. We can place these items *dispersedly* within the list, for example, by randomly positioning the diverse items at different places in the list. On the other hand, the diverse items can be placed together in one *block* in the list. A block means that one section of the recommendation list contains only diverse items. Users may perceive the recommendation list with a block of diverse items to be more diversified than a list with dispersedly placed diverse items since it can be easier for users to discover the block of diverse items. Furthermore, the position of the diverse items may also affect a recommender's overall perceived quality. For example, if the diverse items are placed together on the top of the list, users may get the impression that the recommender's predicting ability is poor and therefore they may lose the trust in the system and stop using it in the future [14].

To the best of our knowledge, how to place diverse items in a recommendation list has not been explored so far in recommender system research. Considering the possible effects of differently positioning the diverse items, we believe that the question of how to arrange the diverse items is an important research topic in recommender systems.

In order to tackle this problem, the aim of this paper is to investigate how to place the diverse items in a recommendation list and analyze the effects of different item placements on the perceived diversity, on serendipity, and on user satisfaction. As a final goal, we want to develop a set of guidelines of how to arrange diverse items so as to improve recommender's overall perceived quality.

The remainder of this paper is organized as follows. In Sect. 2 we propose a set of hypotheses about the placement of diverse recommendations and their potential effects. In order to validate the hypotheses, in Sect. 3, we design an experiment to study the effects of the different placements of the diverse items. Next, we carry out a data analysis and summarize our results in Sect. 4. We conclude this paper by discussing our findings and providing indications how to better arrange the items in a recommendation list.

2 Research Hypotheses

Sakai [17] pointed out that balancing relevance and diversity has been considered as a challenge in document retrieval [5]. This trade-off has been also noticed in the recommender system community. Adomavicius and Kwon [1] stated that increasing diversity in a recommender system can result in decreasing its accuracy and vice versa. Thus a number of recommender algorithms focus on combining diversity and accuracy [16, 20] or increasing diversity with a minimal loss of accuracy [1, 18, 19].

The concept of diversity used in the papers above refers to inherent diversity, which is often measured by the dissimilarity between all pairs of recommended items. Therefore, inherent diversity does not depend on the order of the items and changing the order of diverse items in a recommendation list will not affect inherent diversity. Ziegler et al. [20] therefore argued that rearranging the positions of the items in a recommendation list would not affect inherent diversity. However, as we discussed in the introduction, it may affect the perceived diversity. Specifically, it might be easier for users to discover diverse items when they are arranged in a block than dispersedly placed. We therefore propose the following hypothesis.

H1: A Recommendation List Containing a Block of Diverse Items is Perceived to be more Diverse than one with Dispersedly Placed Diverse Items. Changing the order of diverse items may also affect the serendipity of a recommendation list. Serendipity is considered to be an important factor to attract users to use recommender systems [8]. McNee et al. [15] propose to define it as the experience by the user who received an unexpected and fortuitous recommendation. Thus serendipity can be measured by the extent to which the recommendations are both attractive and surprising to the user [10]. Moreover, Ge et al. [8] found two essential aspects of serendipity: unexpectedness and usefulness. While unexpected recommendations refer to those recommendations that are significantly distant from the user's expectations, usefulness means the highest level of utility to the user. Diverse items are considered to play an important role in generating unexpected recommendations [3].

Intuitively, we assume that users are to some extent surprised when they see diverse recommendations. For example, users may be surprised when seeing a romantic movie within a list of action movie recommendations. Thus, if several diverse items are dispersedly placed in the recommendation list, users can regularly find unexpected items and may experience more "surprise times" than in the case that the diverse items are placed together in a block. We therefore establish hypothesis H2 as follows.

H2: A Recommendation List with Dispersedly Placed Diverse Items is Perceived to be more Unexpected than the one Containing a Block of Diverse Items. Our review above indicates that previous research has realized the potential value of diversity and serendipity in recommender systems. Adomavicius and Kwon [2] argue that increasing diversity can lead to an increase in sales diversity and user satisfaction. Also, as Ge et al. [8] discussed, surprising and serendipitous recommendations can increase the user's interest of using a recommender system, and in turn lead to higher user satisfaction. Therefore maintaining a certain level of diversity and serendipity in a recommendation list can improve user satisfaction. According to the discussion when developing hypothesis H1 and H2, diverse items that are arranged in a block presumably will result in higher diversity, whereas diverse items that are dispersedly arranged will presumably increase serendipity. Increasing either diversity or serendipity can lead to a higher level of user satisfaction. We therefore propose a *null hypothesis,* H3, as follows.

H3: A Recommendation List Containing a Block of Diverse Items can Result in the same user Satisfaction with a Recommendation List with Dispersedly Placed Diverse Items. Overall, our three hypotheses are proposed based on a literature review and our intuitive conjectures. In order to test the proposed hypothesis, we designed an experiment to empirically analyze the effects of different placements of diverse items, which we describe in the next section. Furthermore, as we are also interested in studying whether the presence of diverse items is beneficial for recommender systems in general, we will include a treatment without diverse items in the experiment.

3 Experiment

In this section, we will present the experimental design and measurement technique used in our study. In addition to studying how to arrange diverse items in a recommendation list, we also study whether and to which extent diverse items influence the user-perceived quality of a recommender system. In our experiment, we employ a *within subjects* design, in which each subject can evaluate and compare all the treatments used in this user study.

Our experiment is implemented as an online website. There are three phases in the experiment. The first phase is to instruct the participants about the different phases of the experiment and how they can complete the experiment. The second phase is that subjects interact with a recommender system, rate items and are presented with movie recommendations. In the recommendation phase, we used four movie genres: action

	Title (Action Movie)	I have watched and also like this movie
1.	Mission: Impossible (1996)	
2.	The Dark Knight (2008)	
3.	Iron Man (2008)	
4.	Terminator 2: Judgment Day (1991)	
5.	The Matrix (1999)	
6.	Spider-Man (2002)	
7.	Braveheart (1995)	
8.	Indiana Jones and the Last Crusade (1989)	
9.	The Transporter (2002)	
10.	Gladiator (2000)	
11.	Sin City (2005)	
12.	Fight Club (1999)	
13.	Casino Royale (2006)	
14.	The Bandit (1996)	
15.	Kill Bill (2008)	
16.	300 (2006)	
17.	Snatch (2000)	
18.	Armageddon (1998)	
19.	Heat (1995)	
20.	Die Hard (1988)	

Fig. 1. Screen 1 - Acquiring user preferences for action movies.

movies, romantic movies, comedy movies and animation movies. For each movie genre, we have developed two web pages. In the first web page, subjects are provided with a list of 20 well-known movies of one specific genre. Figure 1 shows an example snapshot in which a list of 20 action movies is presented to the subjects.

The subjects will be asked to check the movies they have watched and also liked. After the subjects finished ticking their preferred movies, they can click on the "Get Recommendations" button to obtain recommendations. Then, on the second web page, shown as Fig. 2, a list of 12 recommended movies is presented to the subject. Three options are offered: "I would like to watch this movie", "I have watched this movie and liked it" and "I have watched this movie but I do not like it". Subjects can tick one of the options to report their opinions towards the recommendations. It is however not mandatory for subjects to tick an option for each recommendation. In order to support the subjects in the decision process, the plot of each recommended movie is also given by the system (refer to Fig. 2). The movie plot and three options were used to let users carefully consider the recommendations.

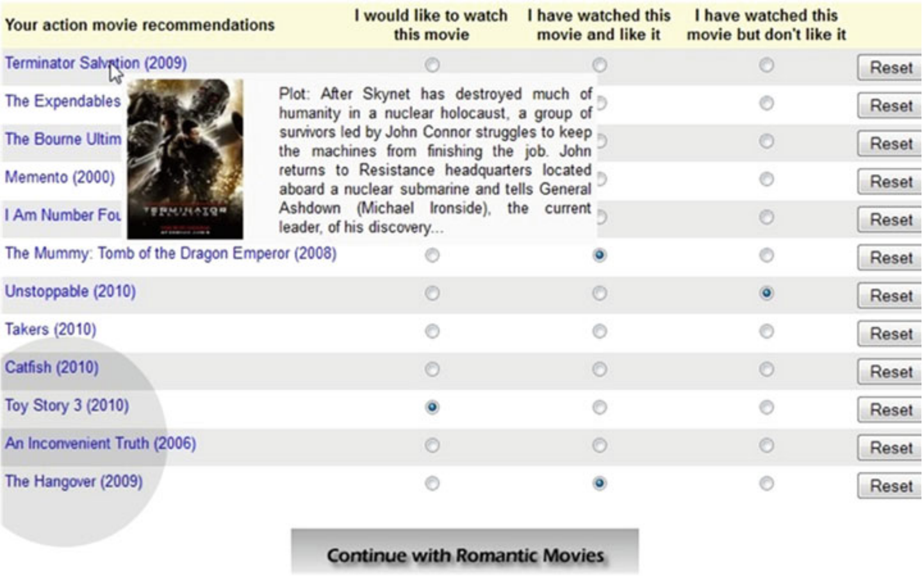


Fig. 2. Screen 2 - Displaying recommended action movie recommendations to users.

It is important to know that in our experiment we do not use any recommender algorithm to compute the recommendations. Instead, we manually create a static list of recommended movies for each genre and present it to users. Therefore each subject will obtain exactly the same set of recommendations. We can thus eliminate possible effects from recommender algorithms. In order to give the user an impression that there is a recommender system running in the background, we not only ask the users about their preferences but also show a message that the recommendations are being computed for two seconds after the subject clicks the "Get Recommendations" button.

In our experiment, we determine diverse movie recommendations based on differences with respect to the movie genre. For example, among the recommended action movies, an animation movie, Toy Story 3, is considered as a diverse item. In the experiment, each recommendation list contains twelve items. Four of them are diverse items. For example, in Fig. 2, there is a list of twelve recommendations. The four diverse recommendations are placed at the bottom of the list. We use a round grey shadow to highlight the four diverse items in Fig. 2. Note that this shadow was of course not visible during the study.

We designed the different placements of diverse items as follows. In the list of action movie recommendations, the four diverse items are organized together in one block at the end of the list. For romantic movie recommendations, the four diverse items are arranged in the middle block of the list. Among the comedy movie recommendations, the four diverse items are respectively placed at the positions 3, 6, 9, and 12. We suppose that diverse items are dispersedly placed in this list. In addition, we use the recommended animation movies as our control group, which contains no diverse items.

After the subjects have gone through every recommendation list, in the last phase they are again presented with all the four manually designed recommendation lists. Subjects are then asked to evaluate each list on a five point Likert scale. The evaluation is based on the following questions, which are designed to test our proposed hypotheses.

- Do you think this recommendation list is diversified?
  (1: not at all, 5: very diversified)
- Does this recommendation list surprise you?
  (1: not at all, 5: very surprised)
- Are you satisfied with the movie recommendations?
  (1: not satisfied, 5: very satisfied)

In the end of the evaluation, the system also displays a textbox where the subjects can leave a feedback regarding the recommendations. After finishing the evaluation, the subject needs to click the "Submit" button to complete the experiment. The whole experiment procedure is supervised in case the subjects need an explanation about system functions or the meanings of some terms. During the experiment there is no interaction between the subjects.

4 Research Result

A total of 52 subjects were involved in the experiment. All the subjects were researchers or students from the computer science department at the Technical University of Dortmund. 35 % of the subjects were female and 65 % were male. The average age of subjects was 29. For each subject, it took on average about 15 min to finish the whole experiment.

As our experiment used a Likert scale, the data collected from the experiment were ordinal data. We therefore choose a non-parametric test to analyze our collected data. Since the same subjects have participated in all the experimental treatments, the Friedman Test is used to test, whether or not there is any difference among the experimental treatments. Once a significant difference is found, the Wilcoxon Signed-Rank Test would be performed to find where the differences actually occur. SPSS 19.0 was used for data analysis and all the tests were done at a 95 % confidence level. We report the analysis results in the following.

As a first step, we performed a Friedman test on perceived diversity. In the test, there are four buckets of data, which are named "Dispersedly", "Bottom", "Middle" and "Without". "Dispersedly", "Bottom" and "Middle" denote recommendation lists where the diverse items are placed dispersedly, at the bottom, or in the middle respectively. "Without" stands for our control group that contains no diverse items. This naming scheme is also applied in all the following tests. The results of the Friedman test for perceived diversity are shown in Table 1.

In Table 1 we can see that there was a significant difference in perceived diversity depending on the placement of diverse items ($\chi^2(3) = 30.890$, $p < 0.05$). This means that different placements of the diverse items significantly affected the perceived diversity of the recommendation list. Thus we arranged the mean ranks in descending

Table 1. Friedman test for perceived diversity.

Mean ranks		Test statistics[a]	
Bottom	3.13	N	52
Dispersedly	2.64	Chi square	30.890
Without	2.56	df	3
Middle	1.68	Asymp. Sig.	.000

order and further performed the Wilcoxon Signed-Rank test to find which group caused the significant difference. The result of the Wilcoxon test for perceived diversity is shown in Table 2.

In order to interpret our Wilcoxon test result, a Bonferroni correction was accordingly applied and thus all the effects are reported at a $p < 0.008$ level of significance.

The result show that it appears that placing the diverse items dispersedly in the recommendation lists is perceived to be more diverse than in the case where the diverse items are placed in the middle ($Z = -3.980$, $p < 0.008$). H1 is therefore rejected and placing the diverse items, for example, in the middle of the recommendation list, does not lead to a higher level of perceived diversity. However, there was no significant difference between placing diverse items dispersedly and at the bottom ($Z = -1.950$, $p = 0.051$). We therefore found that a recommendation list with dispersedly placed diverse items can achieve equal or higher perceived diversity than the one containing a block of diverse items.

Regarding the issue of whether or not including diverse items will increase the perceived diversity, our analysis showed that including diverse items in a recommendation list can both increase and sometimes even decrease the perceived diversity. It depends on how to arrange the diverse items. If the diverse items are placed together in the bottom of a list, the perceived diversity is significantly higher than the list without diverse items ($Z = -2.856$, $p = 0.004$). However, when we place the diverse items in the middle of the recommendation list, the list's perceived diversity is even significantly lower than the one without diverse items ($Z = -3.541$, $p < 0.008$). One possible explanation is that users simply stop inspecting the recommendation list further when they encounter diverse and seemingly non-relevant items in the middle. However, in situations without such diverse elements users may have gone through the whole list. Given the larger number of considered items, they may have found the list generally to be more diverse than the one with diverse items placed in the middle.

In order to examine H2, we performed a Friedman test on the perceived surprise level. The result of the analysis is shown in Table 3.

Table 3 shows that there was a significant difference among the four experimental treatments ($\chi^2(3) = 8.817$, $p = 0.032$), indicating that different placements of diverse items perform differently in surprising the users. Therefore we further used the Wilcoxon Signed-Rank test to find the details regarding this significant difference. The result of this Wilcoxon test is shown in Table 4.

Table 2. Wilcoxon Signed-Rank test for perceived diversity.

	Dispersedly and bottom	Middle and bottom	Without and bottom
Z	−1.950[a]	−4.295[a]	−2.856[a]
Asymp. Sig.	.051	.000	.004
	Middle and dispersedly	Without and dispersedly	Without and middle
Z	−3.980[a]	−.557[a]	−3.541[b]
Asymp. Sig.	.000	.577	.000

[a] *Based on negative ranks*
[b] *Based on positive ranks*

Similar to the analysis for perceived diversity, the Wilcoxon Test was conducted with a Bonferroni correction, resulting in a significance level at $p < 0.008$. The analysis shows that placing the diverse items in a recommendation list dispersedly can lead to a higher surprise level than the in the case where the diverse items are placed in the middle of the list ($Z = -2.755$, $p = 0.006$). There was no significant difference in surprising users when the diverse items are placed dispersedly or at the bottom ($Z = -0.426$, $p = 0.670$). Therefore H2 is partially supported. Interestingly, we found that including diverse items does not significantly increase the surprise level of the recommendation list. This indicates that including diverse items in a recommendation list to the extent we did in our experiment will not increase the surprise level independent of the placement of these items.

Finally, we carried out a Friedman test on user satisfaction. The analysis result can be found in Table 5.

Surprisingly, we found no significant differences among the four experimental treatments ($\chi^2(3) = 3.359$, $p = 0.340$). This indicates that placing the diverse items in a recommendation list dispersedly, at the bottom, in the middle or without diverse items results in the same level of user satisfaction. We therefore fail to reject the null hypothesis H3. That means we found there exists the possibility that all of our experimental treatments result in the same level of user satisfaction. Because there is no significant difference found in the Friedman test, there is no need to carry out the Wilcoxon Signed-Rank test for user satisfaction. As a practical consequence, we are able to add a certain number of diverse items in the recommendation list without hurting user satisfaction. This implies that in practice we can add some extra items to promote certain products or increase sales diversity.

Table 3. Friedman test for the surprise level of the recommendation list.

Mean ranks		Test statistics[a]	
Dispersedly	2.83	N	52
Bottom	2.58	Chi square	8.817
Without	2.53	df	3
Middle	2.06	Asymp. Sig.	.032

Table 4. Wilcoxon Signed-rank test for user satisfaction.

	Dispersedly and bottom	Middle and bottom	Without and bottom
Z	$-.426^{a}$	-2.240^{b}	$-.906^{b}$
Asymp. Sig.	.670	.025	.365
	Middle and dispersedly	Without and dispersedly	Without and middle
Z	-2.755^{b}	-1.271^{b}	-2.462^{a}
Asymp. Sig.	.006	.204	.014

[a] *Based on negative ranks*
[b] *Based on positive ranks*

Table 5. Friedman test for user satisfaction.

Mean ranks		Test statistics[a]	
Without	2.68	N	52
Middle	2.61	Chi square	3.359
Dispersedly	2.50	df	3
Bottom	2.21	Asymp. Sig.	.340

5 Discussion and Implication

A number of algorithms have been proposed to increase diversity or generate diverse items in the recommendation list [19, 20]. However, the issue of how to place the diverse items is still not in the focus of recommender system research. We propose in this work that the question of how to place diverse items is an important issue because differently placing the diverse items can affect perceived diversity and the level of serendipity. Based on our findings, if the goal of recommender systems is to increase the perceived diversity, we suggest positioning the diverse items dispersedly or together in the bottom of the list. It is also important to note that placing the diverse items in the middle of the recommendation list may even reduce the perceived diversity. Furthermore, as we can use the placement of the diverse items to control the perceived diversity, our result might be used to manipulate perceived diversity in future experiment such as in factorial design.

Additionally, we found that in the movie domain including a certain amount of diverse items in the recommendation list does not surprise the users too much. When investigating the role of serendipity in recommender systems, we therefore suggest that further studies should focus on the cross-domain product recommendations. Also, the possibility of improving serendipity might be increased when recommending products from different domains.

A number of studies are based on the assumption that increasing diversity will lead to higher user satisfaction. We therefore tried to analyze whether increasing diversity results in higher user satisfaction. However, we found that there was no significant difference between the groups that received diverse recommendations and the group whose list was more monotonous. One possible explanation is that in the movie domain users usually have a strong or relatively fixed movie preference. Therefore the diverse movies might have been of limited interest to the users. In other domains such as tourism, users might however be interested to see quite different travel destinations. Thus we argue that this can be seen as a domain specific problem and our conclusions are limited to the movie domain.

While we see our work as a further step toward a better understanding of the role of diversity and serendipity of recommendation lists, we are aware of some limitations of our work. First, there might be an effect related to the different movie genres in the experiment. Different movie genres might for example influence the user's evaluation of the system. In order to minimize the effect of different genres, we clearly instructed the subjects that in the experiment the four movie genres are four different scenarios. In a future study, we will further improve the design of the experiment and focus on a single movie genre so as to eliminate the effects of genres. Second, user preference is an external factor that may influence the experiment. User satisfaction might not only depend on the position of diverse item, but also on their personal preference. We tried to avoid this influence by using only very popular and well-known movies in the experiment. Note that users have selected the movies they have watched and also liked in the experiment. Considering this data, we have excluded the subjects with strong movie preferences. In the future, we will further conduct an experiment with the subjects who have similar movie preferences. Lastly, note that in our study each subject obtained exactly the same set of recommendations in order to eliminate possible effects from recommender algorithms. However, one can also plan experiments with a real recommender system which takes user preferences into account to finally decide which items will be perceived as interesting or diverse by the users, independent from the movie genre. We believe that if the recommender algorithm is fixed for all users and enough preference data is available from the users the possible effects from recommender algorithms can be kept under control.

In our future work we plan to conduct a broader study on the positioning effects of diverse items which involves more participants from different demographic user groups. Since our empirical study only focused on the movie domain, which suffers from being subjective in nature, we also aim to test whether our results can be reproduced in other domains, in particular in those, were explicit user preferences or constraints have to be observed, e.g., the recommendations of digital cameras.

As for the evaluation and measurement method, we plan to automatically collect more types of information during our future experiments. Beside the use of post-study surveys, we assume that user actions such as viewing the details of a diverse item during the experiment as well as viewing times can serve as indicators of the level of interestingness or surprise of a recommended item.

References

1. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Trans. Knowl. Data Eng. **99**, 1–15 (2011)
2. Adomavicius, G., Kwon, Y.: Maximizing aggregate recommendation diversity: a graph-theoretic approach. In: Proceedings of Workshop on Novelty and Diversity in Recommender Systems, Chicago, Illinois, USA, pp. 3–10 (2011b)
3. Adamopoulos, P., and Tuzhilin, A.: On unexpectedness in recommender systems: or how to expect the unexpected. In: Proceedings of Workshop on Novelty and Diversity in Recommender Systems, Chicago, Illinois, USA (2011)
4. Castells, P., Vargas, S., Wang, J.: Novelty and diversity metrics for recommender systems: choice, discovery and relevance. In: Proceedings of International Workshop on Diversity in Document Retrieval, Dublin, Ireland, pp. 29–37 (2011)
5. Clarke, C. L. A., Craswell, N., Soboroff, I. and Ashkan, A.: A comparative analysis of cascade measures for novelty and diversity. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, pp. 75–84 (2011)
6. Dias, M. B., Locher, D., Li, M., El-Deredy, W., Lisboa, P. J.: The value of personalised recommender systems to e-business: a case study. In: Proceedings of the 2008 ACM Conference on Recommender Systems, Lausanne, Switzerland, pp. 291–294 (2008)
7. Fleder, D., Hosanagar, K.: Recommender systems and their impact on sales diversity. In: Proceedings of the 8th ACM Conference on Electronic Commerce, San Diego, CA, USA, pp. 192–199 (2007)
8. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: Proceedings of the Fourth ACM Conference on Recommender Systems, New York, pp. 257–260 (2010)
9. Gedikli, F., Ge, M., Jannach, D.: Understanding recommendations by reading the clouds. In: Huemer, C., Setzer, T. (eds.) EC-Web 2011. LNBIP, vol. 85, pp. 196–208. Springer, Heidelberg (2011)
10. Herlocker, L., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Trans. Info. Syst. **22**(1), 5–53 (2004)
11. Jannach, D., Hegelich, K.: A case study on the effectiveness of recommendations in the mobile Internet, pp. 205–208. ACM Conference on Recommender Systems, New York (2009)
12. Jannach, D., Zanker, M., Felfernig, A., Friedrich G., Recommender systems: an Introduction, Cambridge University Press (2010)
13. Jannach, D., Zanker, M., Ge, M., Gröning, M.: Recommender Systems in Computer Science and Information Systems – A Landscape of Research. In: Huemer, C., Lops, P. (eds.) EC-Web 2012. LNBIP, vol. 123, pp. 76–87. Springer, Heidelberg (2012)
14. Lathia, N., Hailes, S., Capra, L., Amatriain, X.: Temporal diversity in recommender systems. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, pp. 210–217 (2010)
15. McNee, S., Riedl, J., Konstan, J.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems. Montréal, Canada, pp. 1097–1101 (2006)
16. Smyth, B., McClave, P.: Similarity vs. Diversity. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 347–361. Springer, Heidelberg (2001)
17. Sakai, T.: Challenges in diversity evaluation. In: Proceedings of International Workshop on Diversity in Document Retrieval. Dublin, Ireland, pp. 1–7 (2011)

18. Zhou, T., Kuscsika, Z., Liua, J., Medoa, M., Wakelinga, J., Zhang. Y., Solving the apparent diversity-accuracy dilemma of recommender systems. In: Proceedings of National Academy of Sciences of the USA. vol. 107 (10), pp. 4511–4515 (2010)
19. Zhang, M., Hurley, N.: Avoiding monotony: improving the diversity of recommendation lists. In: Proceedings of the 2nd ACM Conference on Recommender Systems, Lausanne, Switzerland, pp. 123–130 (2008)
20. Ziegler, C., McNee, S., Konstan, J., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th World Wide Web Conference. Chiba, Japan, pp. 22–32 (2005)
21. Zanker, M., Bricman, M., Gordea, S., Jannach, D., Jessenitschnig, M.: Persuasive online-selling in quality and taste domains. In: Bauknecht, K., Pröll, B., Werthner, H. (eds.) EC-Web 2006. LNCS, vol. 4082, pp. 51–60. Springer, Heidelberg (2006)

Time Efficiency of Point-of-Sale Payment Methods: Empirical Results for Cash, Cards and Mobile Payments

Michal Polasik[1(✉)], Jakub Górka[2], Gracjan Wilczewski[1], Janusz Kunkowski[1], Karolina Przenajkowska[1], and Natalia Tetkowska[1]

[1]Faculty of Economic Sciences and Management, Nicolaus Copernicus University, Gagarina 13A, Torun, Poland
michal.polasik@umk.pl, gracjanw@wp.pl
janusz.kunkowski@umk.pl, caroline@doktorant.umk.pl
natalia.tetkowska@gmail.com
[2]Faculty of Management, University of Warsaw, Szturmowa 1/3, Warsaw, Poland
jgorka@jgorka.pl

Abstract. We propose a novel approach for the time efficiency study of payment process at Point-Of-Sale (POS). A wide range of payment methods from cash and standard cards to contactless cards, RFID stickers and mobile payments (NFC and remote) was analysed. Transactions were timed by means of digital chronography of video material recorded in the biggest chain of convenience stores in Poland. Our results confirm that cash is a significantly faster payment method than traditional payment card with a magnetic stripe or EMV chip. However, the innovative payment methods, such as contactless cards and NFC mobile payments, are competitive to cash in terms of time efficiency. Contactless cards used in offline mode and without printing paper slips are the first popular electronic payment method in history faster than cash. Our results could be applied to optimise the payment process at POS as well as to develop innovative and efficient payment solutions.

Keywords: Retail payments · Tender time · Credit & debit cards
Contactless cards · NFC · Mobile payments

1 Introduction

The modern retail payment market offers a wide range of payment methods which may be used at the Points-Of-Sale (POS), such as cash, cards, cheques or currently even mobile devices. Payments at physical POS have an important impact on the entire economy, as they are a daily activity of consumers. Consumer's choice of a payment method is influenced by numerous factors which *inter alia* comprise its pecuniary cost

---

The original version of this chapter was revised: It has been changed to open access under a CC BY 4.0 license, and the copyright holder has been updated as 'The Author(s)'. The correction to this chapter is available at https://doi.org/10.1007/978-3-642-40654-6_29

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 306–320, 2013.
DOI: 10.1007/978-3-642-40654-6_19,

of use, convenience and certainty of acceptance [2, 9]. Individuals must have compelling reasons to change their payment habits. Otherwise they keep using means of payment they are most familiar with [3]. Consumer satisfaction from the purchase depends heavily on the time of queuing and the time of undertaking a transaction at the counter [1, 24]. As a consequence of strong competition in retail trade, aspects of payment costs, sales organisation and the length of queues have great impact on profitability of merchants business and their market success. Modelling the flow of customers and the time spent by them while paying, may result in decreasing queues which is advantageous for both merchants and consumers.

The main aim of our study was to measure the time efficiency of different payment instruments used at physical POS in Poland. The time efficiency of payment methods understood as the speed of the transaction process has been evaluated on the basis of empirical research that was conducted in the chain of convenience stores. We were willing to verify whether new payment solutions like contactless (proximity) cards, RFID stickers or mobile payments – the remote system and NFC mobile phones – were faster in a transaction process than traditional payment methods – cash and standard debit and credit cards.

To the best of our knowledge, this is the first study in the world to gauge payment process time components of a wide range of payment methods used at POS.

2 Importance of Transaction Speed at the Check-Out

The speed of the payment transaction process is of major importance for the efficiency of a given instrument. Time of servicing a payment transaction by a seller should be treated as a part of merchants costs. Garcia-Swartz et al. [7, 8] label these costs as 'tender time costs'. They are a part of front-office costs and are usually translated to staffing costs at the average wage rate for shop workers [4]. Thus, the time of a transaction at the check-out constitutes an important part of merchants costs. Moreover, time spent on paying and queuing implies consumers cost. A reduction of queue lines can therefore decrease consumers costs of payment [4]. At the same time – from the merchants' viewpoint – reducing lines in shops through the shortening of the payment process may boost sales, because fewer clients will withdraw from the purchase. This element gains significance because consumers become more demanding. Research published by Barclays Bank and Populus Ltd in 2010 shows that customers are unwilling to wait in line to pay for their shopping. Two-fifths of the shoppers refuse to wait in queue for more than two minutes while two-thirds regularly abandon purchase [1]. It shall be noticed that the importance of the transaction speed varies depending on the sector – it is of key importance during mass events, in public transport or fast-food chains and less in luxury boutiques or restaurants.

In addition, speed can contribute to the market success of a new payment instrument, therefore this feature can be vital for payment providers. Studies conducted worldwide have shown that the speed of a transaction, determining time spent at the counter, is one of the most significant factors determining the choice of a payment instrument [11–13, 18, 19, 22, 23, 25]. In particular young clients negatively react on longer executing times of executing a payment [3]. This suggests that instruments requiring more effort on the consumer's part have little chance of gaining popularity.

Due to their physical and technological features, which are the most important for the presented study, payment instruments can be divided into three groups: (a) paper instruments: cash (banknotes and coins) and cheques, (b) cards: debit, credit, pre-paid and e-purses (electronic money), and (c) mobile devices working in remote schemes and proximity schemes. The usage of these instruments is much diversified across countries. However, even in countries with several decades of experience in card use, cash still remains the main form of payment. According to McKinsey [15], cash was used in 70–93 % of the total number of retail transactions concluded by households in the most important western economies in 2005.

Recently many new solutions have been introduced to the market, which can become competitors of cash at physical POS. One of the most important innovations are contactless cards. Contactless payments are based on an extension of RFID technology (Radio Frequency Identification) enabling remote reading of integrated circuits via radio waves [10]. The first application of this technology for payments took place in 1997 for the Hong Kong's public transport network [14], and first contactless payment cards for more general banking purposes, MasterCard *PayPass*, were issued in United States in 2002 [5]. In Poland, this technology was pioneered by Bank Zachodni WBK SA in December 2007. In 2011 Poland became one of the leading contactless market in Europe. Most of the contactless card are issued in traditional form, however other forms, like RFID stickers for mobile phones, wristwatches or keyfobs, are also applied to a limited extent and they operate similarly to contactless cards. The more advanced contactless mobile payment technology, i.e. NFC (Near Field Communication is an expansion of RFID technology, complimentary to contactless card; see [10]) can also work similar to a contactless payment card. However, NFC has much more features based on mobile device, and one of them is optional activation using a PIN code. Mobile NFC payments and contactless cards use the same contactless POS terminals network, what generates the synergic effect. The main obstacles for market development is a very limited number of mobile devices equipped with NFC technology. As a result, NFC mobile payments have not been commercially deployed on a large scale except Japan and South Korea.

An alternative solution, which can be used in POS transactions, are remote mobile payment systems. These systems are based mostly on universal communication through the GSM mobile network (most often through SMS or USSD sessions) or mobile Internet. The versatility of these solutions results from that the acceptance of payment is possible through an adapted POS terminal, a WEB-terminal (online payments), or a seller's handset. There are many more or less successful remote mobile payment schemes operating in the world today, a majority of which use the GSM technology and/or an installed application. Most of them are domestic systems, including Obopay and PayPal (USA), mChek and PayMate (India), M-PESA (Kenya), MoneyBox (Nigeria), and mPay (Poland) to mention the popular ones. Some of the above mentioned payment innovations could have potential in accelerating the payment process.

Several studies on the speed of payment transaction with different instruments have already been undertaken. Results of the most significant or most widely cited, are presented in Table 1. There are quite significant discrepancies in transaction times among studies due to methodology used. Klee [12, 13] in the USA and probably also the Belgian study [20] focused on the time of a payment transaction comprising ringing up items. The Dutch estimates did not include this stage. Transaction times for Netherlands are therefore shorter across all payment instruments. With the exception of the Smart Card Alliance approximate data, all other studies point out that cash is faster than traditional payment cards. According to Dutch and Belgian studies the fastest payment instrument is an electronic purse [4, 20]. Note that none of the analyses split the transaction times into stages. This subject thus requires more detailed investigation. The presented study sheds light on the duration of particular stages of payment transaction for many payment instruments, not only traditional ones but also new and innovative.

The Smart Card Alliance study suggests that the new contactless technology is advantageous, because proximity cards have potential to facilitate much faster transactions at POS than cash and traditional cards. Owing to the fact that the contactless technology develops quickly in the USA, many Asian countries and lately also in Europe [19], verifying the time of the payment transaction performed with proximity cards was therefore one of the goals of this research.

3 Research Methodology and Obtained Data

Empirical research was based on chronography of the purchase transaction process at cash registers with the help of video monitoring during regular work hours of shops and test transactions after closing time. In this work we will mainly focus on the results

Table 1. Overview of estimates for the duration of the payment process by instrument [in seconds].

Instrument	USA[a]	Belgium[b]	Netherlands[c]	USA[d]
Cash	34.75	32	19	33.7
Credit card (traditional contact technology)	55.13	56	28	26.7
Debit Card (traditional contact technology)	50.43	39	26	
E-purse (contact microchip technology)	–	20	14	–
Contactless card (RFID technology in on-line mode)	–	–	–	12.5
Checks	77.53	–	–	–

[a] [12]: Data represent the estimated minimum time at the checkout counter for four items purchased at a supermarket (see [3]). The time for the transaction is the "ring time", which is calculated as the number of seconds between the first item crossing the scanner to the close of the cashier's drawer – the amount of time the cashier spends ringing up the transaction (see [12] for a detailed study of consumer response to time at the checkout counter).

[b] [20]: The Belgian Federation of Distributors (FEDIS) carried out a study of the time taken to pay for purchases with different payment methods - cash, electronic purses, debit cards and credit cards – in about 10 distributors. It weighted the average settlement times for the different distributors according to their turnover.

[c] [4]: The time of processing the payments is defined as the length of time between the moment the customer has been informed of the balance due and the moment the sales slip, change etc. has been handed to the customer. The time spent by a cash-register attendant ringing up individual items is not included, since in the study these activities, though necessary to complete a transaction, are not treated as payment activities. In the Cost Survey by the Nederlandsche Bank it was assumed that the duration of the payment process are typical for POS.

[d] [21]: Estimates were performed for the transactions at pharmacies based on data from American Express and CVS/pharmacy. This data should be treated only as approximate.

concerning duration of payment transactions at the checkout. Our research covered all payment methods available in Poland that can be used at POS as well as new solutions that stand a great chance to become popular on the market. Thus, we have not only tested basic payment instruments but also evaluated the uncommon innovative payment methods. The final set of payment method was following: cash, traditional card with signature, traditional card with PIN code, contactless card in online and offline mode, RFID mobile sticker, NFC mobile payment with PIN code, and remote mobile payment (exampled by mPay[1]).

In order to provide full complexity of the payment process we have taken under study three modes of measurement: (1) regular flow of customers; (2) mystery client tests (testers using given payment instruments queuing among regular customers); (3) closed tests (testers only, after shop closure). In the process of a payment we investigated the time of queuing and the time of a purchase payment gauged from the moment of presenting merchandise by a client to a cashier until consummating the payment and handing out receipts. Every payment method has its own specific series of time components, although some of them may be the same – like for example the scanning of items. Nevertheless, the time components vary considerably among payment methods. In the case of cash we have: scanning of items, taking out and counting money by a consumer, taking money by a shop assistant, giving back the change with receipts, whereas in case of a payment card verified by PIN we can observe the following stages: scanning of items, card preparation by the consumer, terminal activation and card handling by the cashier, entering the PIN code on a PIN-pad by the

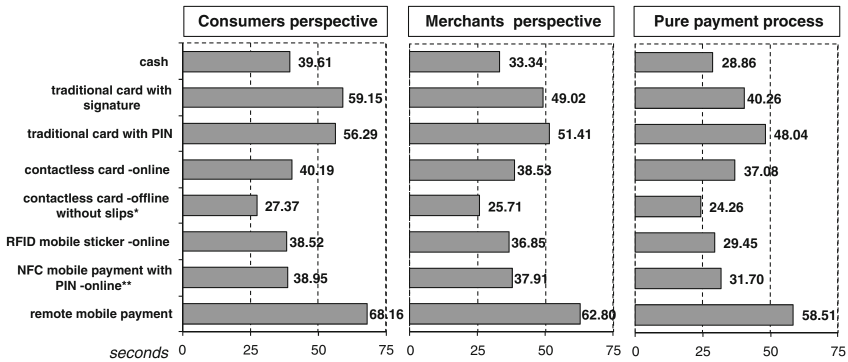


Fig. 1. Average duration of a purchase transaction by payment methods [in seconds]. * Due to the lack of a sufficient number of observations for contactless cards in an offline mode without printing paper slips, the time for this payment method was estimated on the basis of simulation procedure. ** By analogy to contactless cards, the NFC mobile payment with PIN in offline mode and without slips printed would be shorter by about 13 s.

[1] mPay is a remote mobile payment system based on telecommunication connections in GSM standard – text USSD or voice IVR. It can be used on any mobile phone without having to install additional software. A transaction is performed by entering text codes on the mobile. mPay mainly works as an electronic purse and payments can be made to merchants as well as to any mobile phone user.

consumer, slip printing and handing it out to the consumer. Every time a given stage had to be precisely defined and measured.

The empirical data for the study was gathered between the 19th and the 21st of November 2009 in Torun, a Polish city, in a number of convenience stores. More than 30 people were directly engaged in the process (testers, pollsters, organisers and technical support). 3,728 different transactions and tens of thousands of particular time elements for all analysed payment instruments were observed.

After the completion of the process of tests and recording the video, the chronography stage began. The team of IT specialists developed a dedicated computer program Chrono-Metrics which facilitated measurement of all transactions and its time components based on the recorded video material. 95 qualified trainees took part in measurement which lasted for more than 2 months. Each transaction in every shop was on average measured by 5 people in order to minimize errors. Eventually the set of raw data was collected and, after controlling errors, average times of transactions of payment instrument were calculated. During this process the average starting and ending time points for all stages of the payment were determined (see Figs. 2 and 3).

One has to be aware of the limitations of the study, that presents results typical for transactions in the Fast Moving Consumer Goods sector. Therefore it is advised to carefully make generalizations of the results for other sectors.

4 Empirical Results

The obtained results confirmed the dominance of cash in retail POS transactions in Poland. In the regular flow of customers, 94.3 % of transactions were made in cash and 5.7 % with standard payment debit and credit cards. All the other payment methods could only be examined during mystery and closed clients' tests. Nevertheless it was confirmed that the structure of POS payments in convenience stores was roughly identical with the structure of payment estimated for all sorts of shops in Poland [16, 18].

The detailed empirical data gathered in the process of chronography allow to conduct an analysis of all time components of the purchase payment process using multiple variable dimensions. In this paper, we decided to concentrate on the comparison of the time process of payments using three approaches which have very important practical value (Fig. 1). The first takes the perspective of a consumer, the second of a merchant and the third is a 'pure' payment process (without non-payment components of the purchase transaction). From the consumer perspective, the transaction starts with the preparation of a payment instrument or with passing items to the salesperson and ends with the moment when the consumer walks away from the counter. In the view of a consumer, the whole time devoted by him to the execution of a payment is important. The chart below (Fig. 1 on the left side) demonstrates the times of payment transactions for different payment methods from the consumer viewpoint. The second chart (Fig. 1 in the middle) presents the merchant perspective – from the moment of a beginning of items' scanning (customer service) to handing out receipts to the customer. For the merchant, the full time of a salesperson's activity is important. The third chart (Fig. 1 on the right side) exhibits the 'pure' payment process, which

Table 2. Summary statistics for the duration of a purchase transaction, by payment method [in seconds].

Payment instrument	Mean	Median	Minimum	Maximum	Standard deviation	Sample size
The consumer perspective						
Cash	39.61	33.32	4.75	244.35	25.35	2577
Traditional card with signature	59.15	49.93	28.01	216.40	36.41	30
Traditional card with PIN	56.29	48.30	29.21	155.18	22.43	186
Contactless card - online	40.19	37.06	21.84	134.73	12.54	146
RFID mobile sticker	38.52	37.43	29.61	52.35	4.84	26
NFC mobile payment with PIN	38.95	39.01	32.64	43.75	3.06	14
Remote mobile payment	68.16	58.76	38.26	165.64	28.44	52
The merchant perspective						
Cash	33.34	27.59	5.35	192.26	20.73	2577
Traditional card with signature	49.02	40.72	24.21	200.43	32.67	30
Traditional card with PIN	51.41	44.48	16.71	150.81	19.96	186
Contactless card - online	38.53	35.95	21.00	130.13	11.63	146
RFID mobile sticker	36.85	36.26	28.51	49.54	4.38	26
NFC mobile payment with PIN	37.91	38.01	31.97	42.18	2.85	14
Remote mobile payment	62.80	54.82	36.29	152.69	24.05	52
The 'pure' payment process						
Cash	28.86	25.24	2.82	158.59	16.08	2577
Traditional card with signature	40.26	36.26	21.60	75.89	13.96	30
Traditional card with PIN	48.04	41.60	15.39	140.03	19.95	186
Contactless card - online	37.08	30.95	14.66	128.11	14.64	146
RFID mobile sticker	29.45	28.70	22.34	35.73	3.49	26
NFC mobile payment with PIN	31.70	30.75	24.58	36.16	2.95	14
Remote mobile payment	56.51	50.31	33.49	146.46	22.41	52

begins with the preparation of a payment instrument and ends with handing out receipts to the customer. The summary statistics for the duration of a purchase transaction by payment methods are presented in Table 2.

It is evident that for all payment instruments the duration of a transaction is the shortest in the third approach. In the merchant perspective approach, times of transactions are a bit longer. They are the longest from the consumer perspective. The differences arise from the fact that moments of start and stop are not the same. It should, however, be kept in mind that the different stages of a transaction process can overlap. For instance, the customer usually starts preparing the payment instrument while items are still being scanned. Therefore it cannot be stated, as other studies suggest (e.g. [4]), that the pure payment process excludes completely the time of ringing up items.

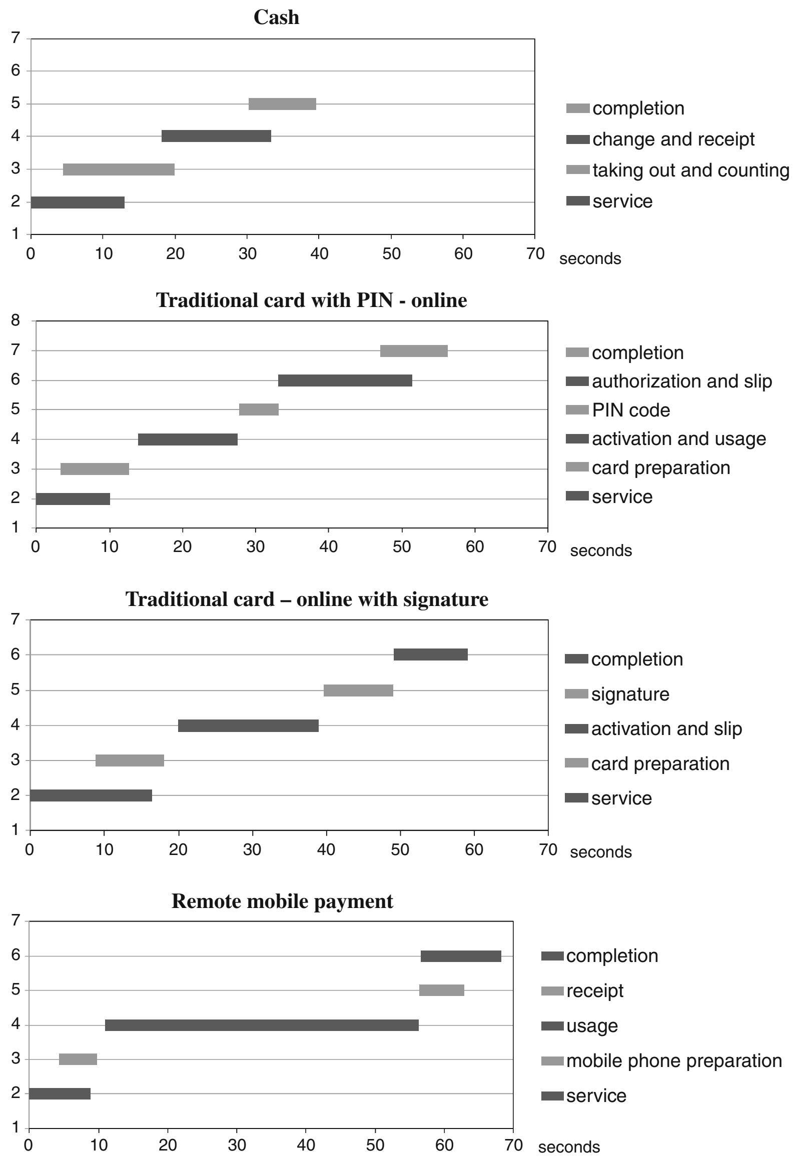


Fig. 2. The stages of the payment process (I).

The merchant perspective approach is the most methodologically comparable to the study by E. Klee [12]. The length of time of the transaction in her work was evaluated on the basis of scanner data, containing the records of a store register receipt with a time stamp, as well as information on the number of items bought, the value of the sale, the number of the store and the payment type used, etc. The results calculated for

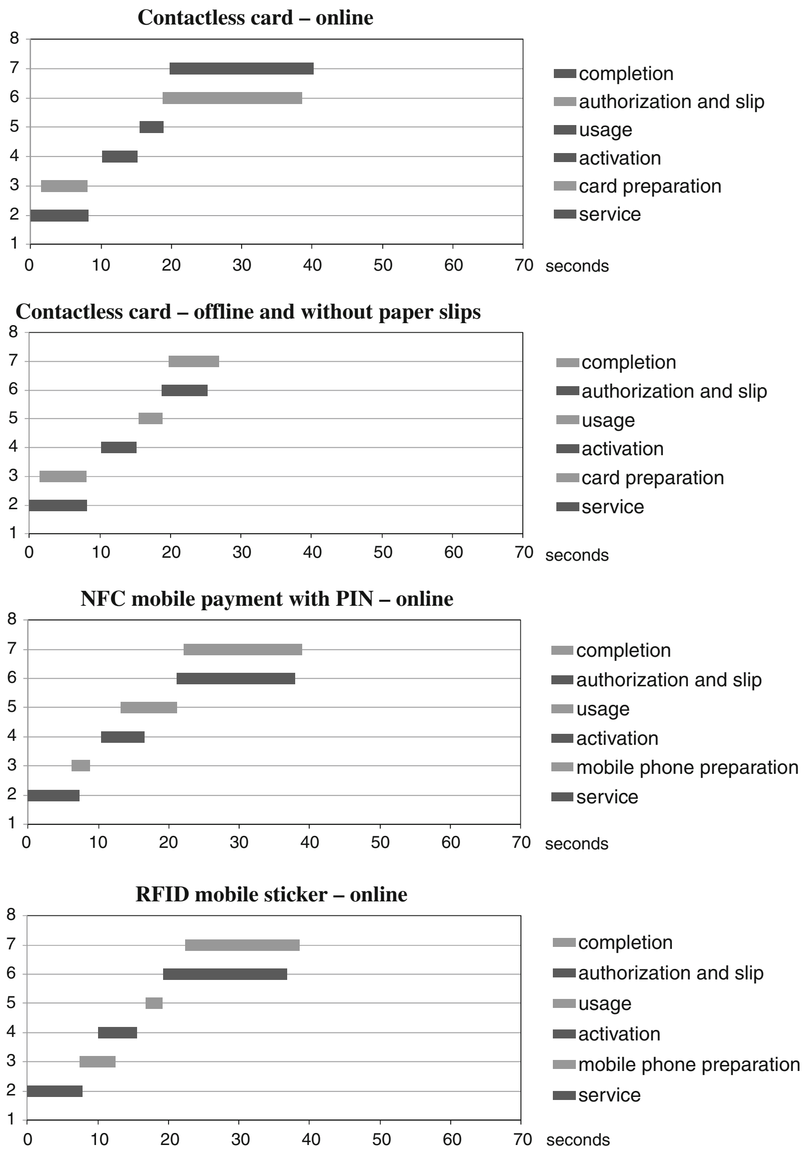


Fig. 3. The stages of the payment process (II).

purchases of four items [3] are comparable to the results presented in our work, as the typical basket of goods acquired in the examined stores consisted of three items. In fact, our results for the merchant perspective (Fig. 1 in the middle) and Klee's results (see Table 1) for the duration of cash and traditional card transactions are very similar.

A first interesting empirical result (Fig. 1) is that in convenience stores cash still stands out as faster than traditional cards (either confirmed by PIN or with signature).

The cash payment is especially short from the perspective of a merchant and in the pure payment process. Usage of traditional contact cards increases the time spent by a consumer at the counter by about 50 %. Such a significant difference in duration of a payment transaction (about 20 s) may also have an effect on the queue when most clients decide to pay with their traditional cards.

From the consumer perspective (Fig. 1), payment by cash and contactless card in online mode (as well as by other RFID instruments) are equally fast.[2] As far as the merchant perspective is concerned, cash payment takes marginally less time. However, when used in online mode contactless cards do not exploit their full technological advantage. In fact, when using contactless cards in offline mode with abandoning slip printing, it appears that transaction with a contactless card lasts significantly less than cash transaction (on average 12.3 s less in the customer perspective). Thus it seems that declarations of card issuers that usage of contactless cards may shorten queues in shops are true (see Table 1). The possibility of cutting down the time of a transaction arises from two sources. The application of offline mode instead of online mode for card payments generates average time savings of 6.61 s. Offline transactions are recommended by payment organisations for contactless cards, and since 2010 most such cards issued in Poland are offline mode enabled. Polish merchants are accustomed to regulations requiring the printing of two paper slips for card transactions. However, we estimated that printing them lasts on average 6.2 s. Consequently, changes of payment organisation rulebooks, which allow not to print paper slips for low value payments, might lead to a further speeding up of contactless card payments. Therefore, we observe a technological breakthrough, because for the first time in history an electronic payment instrument that turns out more time-efficient at the POS than cash, is being issued on a mass scale.

Interesting conclusions pertain to proximity mobile payment such as RFID stickers or NFC which undoubtedly are time efficient (Fig. 1). Some clients can even pay more quickly with their mobile phones than with contactless cards taken out of wallets. These contactless solutions are technologically mature and based on international standards. They gained quite a big popularity in some countries, especially in Japan and the USA [6]. These factors greatly increase the probability of their further development. Popularisation of remote type mobile payments at POS transactions seems to be more difficult. Such payments are characterized by a wider functionality than proximity type mobile payments because apart from POS payments they can be used for transactions on the Internet, for bill payments, parking meters or purchase of tickets in a mass transit communication system. However, due to a considerable number of remote mobile payment systems operating in the world, they are not compatible with each other [17]. This is one of the reasons why they have not become common.

An additional barrier for market success of remote mobile payments is a necessity for consumers to type some information on the phone's keypad and this might be time

[2] Differences in the average duration of transaction between cash, contactless card (online mode), RFID mobile sticker and NFC mobile payment with PIN (Fig. 1) are statistically insignificant from the consumer perspective (see Appendix).

consuming. Indeed, the Polish domestic remote mobile payment system proved to be the slowest payment method out of the analysed (Fig. 1). Nevertheless it must be emphasised that remote mobile payments were only slightly slower than payments with traditional cards and there are potential ways that – if properly implemented – may successfully speed up the transaction process (e.g. by predefined codes).

In order to assess the reasons for differences in transaction time and to find bottlenecks that slow down the process, a more detailed analysis is needed. The process of payment was divided into several stages, which were measured separately using the video chronography method (see Chap. 4). Each stage of the transaction is processed by either a consumer, salesperson or with the use of a technical device. The set of stages differs for every payment method (Figs. 2 and 3). For each stage the moment of its beginning and completion is marked. It is crucial to notice that many activities related to the payment process can be undertaken at the same time, by the client, salesperson or processed by the information system, which shortens the duration of transaction.

The first payment method analyzed was cash (Fig. 2). The stage that delays the payment process of cash is handing over the change and receipt. Although taking out and counting money by the client also lasts long, it mostly runs at the same time as the service, which is the process of charging for the purchased items by the salesperson. When we compare cash payment with slower transactions by traditional card with PIN code, a first important observation is the long duration of: terminal activation, entering the PIN code, the authorization and printing the slip. Due to the fact that those stages are performed one after another by the salesperson and the client, and with the use of a payment terminal, it is impossible to run them parallel, which extends the whole transaction. Where the traditional payment card with the signature is concerned, those middle stages are shortened, because there is no pause for entering the PIN code (the stage lasts constantly from the terminal's activation through the online client's bank account authorization until the printing of the slips). However, at the end the client is involved in signing the slip, which takes more time than the process of entering the PIN code. Moreover, the signing process does not let consumer to prepare to leave the counter with the purchased goods. As a result, sign-based transactions last a little bit longer than transactions with a PIN code.

When contactless cards are considered, the payment process runs in an entirely different way than with traditional payment cards. The stages of terminal activation and tapping the contactless card are extremely short. For contactless cards in online mode, the longest stages are authorization and printing of the two slips. Those stages can radically be shortened when the offline mode is used (Fig. 3). Despite the fact that consumers are satisfied from using contactless cards in offline as well as online mode, it seems that from an economic point of view in order to shorten the payment process using offline mode is reasonable.

The results of our study also enable a comparison between contactless cards and NFC mobile payments (Fig. 3). The NFC payment method requires an activation of payment functions with the use of a PIN code before the transaction can be made. We observed that users take out their mobile phone much faster than their payment card. At the same time, entering the PIN code before using a mobile phone at the terminal, does not extend the time of a transaction, because users enter this code while waiting for the activation of the terminal by the salesperson. In conclusion, the NFC mobile payments are as time effective as those where contactless cards are used.

The situation is different for the remote mobile payments model. In the study, we examined one system operating in Poland branded mPay (Fig. 2). In this case the stage of using the payment instrument lasts fairly long. The system requires the payee to enter certain codes on the mobile phone related to the type of transaction, number of the store and the amount to be paid. Authorizing is done by a PIN number. The remote mobile payment system tested turned out to be the slowest payment method of all, even though the difference between that payment method and the traditional payment card with PIN code was rather small. There are also psychological disadvantages: the consumer, while using this method, is watched by the salesperson and other impatient clients waiting in the queue. This was the reason why users taking part in tests evaluated the remote mobile payments lower than the cards in terms of satisfaction.

5 Summary and Conclusions

The presented results provide novel insights into the important problem of the time efficiency of payment methods at physical POS. It has also revealed that, until recently, cash was the fastest payment method available at POS. As the results of the earlier studies in the other countries show, speed of a transaction is one of the most significant factors determining the choice of a payment instrument by consumer. Therefore, the time efficiency of cash, proved in our study, is probably one of the reasons why people use it so often. Traditional payment cards equipped with a magnetic stripe or EMV chip are much slower. The average time difference between cash and cards is fairly large, about 20 s, and amounts to half of the entire duration of a cash payment transaction. It suggests that frequent use of traditional cards by customers lengthens queues and may cause cost increases for merchants, with additional work for cashiers and risk of losing sales.

Moreover, our results have also allowed verifying the time efficiency of new payment solutions, which are expected to revolutionise the payment market. The most widely known were the contactless cards promoted by payment organisations MasterCard and Visa. It turned out that thanks to the innovative contactless technology, in terms of transaction speed, such payment instruments are likely to become true competitors of cash. Contactless cards and other types of payments devices – mobile stickers and NFC mobile payments – were almost as fast as cash, even despite processing in an online mode and with printing paper slips. After removal of these restrictions and operating in offline mode, transactions with contactless cards can be even quicker than cash. This may be considered a technological breakthrough in the evolution of the means of payment. For the first time in history, an electronic payment instrument, being issued on a mass scale, turns out to be more time efficient at POS than cash. Furthermore, the speed and convenience of contactless payments make them attractive for merchants and customers in low-value payments.

It seems that the development of proximity mobile payment, including NFC payment with PIN code, can be expected. These payment solutions have proved to be highly time efficient and were well received by consumers during the study. However, because of limited availability of mobile phones equipped with NFC technology, the RFID mobile stickers and other contactless payment gadgets have the potential to be part of the first wave of the contactless payment revolution. The time efficiency together with the strong support from the side of international payment card organisations suggest that the dynamic development of contactless mobile payments around the world is highly probable. In contrast, remote mobile payment schemes are rather slow in POS purchases. The execution of transactions in text mode seems to be a major barrier to their application at physical POS. Due to a lower time efficiency, it seems that remote mobile payments, at the current stage of their development, should rather be directed to other payment market segments (e.g. e-commerce, remittance, bill payments, public transport), where their additional functionality may be fully used.

The results presented in the paper are important from a scientific, policy as well as a business viewpoint. The data enable not only to assess the average time of a purchase transaction but also to determine the influence of payment methods on the queuing time and costs borne by merchants. The results of the study allow also to asses consumers' costs of queuing and finally more precise estimation of the general social costs of payment methods. The comparison of many payment instruments gives valuable knowledge that can be useful in preparing strategies for promoting efficient payment instruments among consumers and merchants, e.g. by central banks or public authorities, therefore they are important for policy reasons. Moreover, the results may be especially important for companies introducing new payment solutions to the market.

The detailed timing study of payment process will allow in the future to build an accurate model of the whole payment process covering customer and salesperson activities, and including other factors such as: the type of terminal or a number of items purchased. We also plan to use the obtained data for modelling the process of queue formation and reduction, as well as for the simulation of functioning of the newly designed payment instruments. It would also be very interesting to extend the study on the other types of POS.

Acknowledgements. This work was supported by the Polish Ministry of Science and Higher Education under Grant No. N N113 308835. We wish to express our appreciation to: Zabka Polska SA, MasterCard Europe, and MCX Systems Ltd. for supporting the data collection at POS. Moreover, we are grateful to 61 students of the Nicolaus Copernicus University in Torun and 45 students of the Faculty of Management, University of Warsaw, for their participation in the study. We would also like to thank Leo Van Hove for helpful comments and suggestions.

Appendix

The results of verification for equality of two expected values of transaction durations for compared payment methods.

Payment methods	Perspective	Traditional card with signature	Traditional card with PIN	Contactless card - online	RFID mobile sticker	NFC mobile payment with PIN	Remote mobile payment
Cash	C	−2.926***	−9.579***	−0.482	1.023	0.698	−7.032***
	M	−7.629***	−11.78***	−4.984***	−3.712***	−5.307***	−8.793***
	P	−4.479***	−12.17***	−5.344***	−0.276	−3.361***	−9.074***
Traditional card with signature	C		0.417	2.817***	3.071***	3.015***	−1.159
	M		−0.977	4.650***	5.517***	5.117***	−3.619***
	P		−2.281**	1.707*	4.281***	3.239***	−4.189***
Traditional card with PIN	C			8.226***	9.296***	9.369***	−2.728***
	M			7.240***	8.461***	8.064***	−3.197***
	P			6.066***	11.148***	9.233***	−2.952***
Contactless card	C				1.186	0.936	−6.731***
	M				1.300	0.508	−7.009***
	P				4.659***	2.648***	−6.492***
RFID mobile sticker	C					−0.343	7.169***
	M					−0.918	−7.551***
	P					−2.538**	−8.842***
NFC mobile payment with PIN	C						7.114***
	M						−7.294***
	P						−7.947***

Notes: "C" stands for "Consumer perspective", "M" stands for "Merchant perspective" and "P" stands for "Pure payment process". *, **, *** denote statistical significance at 10 %, 5 % and 1 %, respectively.

References

1. Barclays Bank: Britain isn't queuing. Barclays Bank Press Releases, August (2010)
2. Bolt, W., Jonker, N., van Renselaar, C.: Incentives at the counter: An empirical analysis of surcharging card payments and payment behaviour in the Netherlands. J. Bank. Finance **34** (8), 1738–1744 (2009)
3. Borzekowski, R., Kiser, E.K.: The choice at the checkout: Quantifying demand across payment institutions. Int. J. Ind. Organ. **26**(4), 889–902 (2008)
4. Brits, H., Winder, C.: Payments are no free lunch. Dutch National Bank Occasional Studies **3**(2), 14 (2005)
5. Capizzi, M.T., Ferguson, R.: Loyalty trends for the twenty-first century. J. Consumer Marketing **22**(2), 72–80 (2005)
6. Eastwood, G.: The Future of Payments. Prepaid cards, Contactless and Mobile Payments, Business Insights, Datamonitor (2008)
7. Garcia-Swartz, D., Hahn, R., Layne-Farrar, A.: The move toward a cashless society: a closer look at payment instrument economics. Rev. Netw. Econ. **5**(2), 175–198 (2006)
8. Garcia-Swartz, D., Hahn, R., Layne-Farrar, A.: The move toward a cashless society: calculating the costs and benefits. Rev. Netw. Econ. **5**(2), 199–228 (2006)

9. Górka, J.: Competitiveness of money forms and payment instruments. CeDeWu, Warsaw, Poland, pp. 1–214 (2009) (in Polish)
10. Hancke, G.: RFID and Contactless Technology. In: Mayes, K.E., Markantonakis, K. (eds.) Smart Cards, Tokens, and Security Applications. Springer, New York (2008)
11. Jonker, N.: Payment instruments as perceived by consumers – results from a household survey. De Economist **155**(3), 271–303 (2007)
12. Klee, E.: Paper or Plastic? The Effect of Time on the use of Checks and Debit Cards at Grocery Stores, Finance and Economics Discussion Series, No. 2006-02. Washington: Board of Governors of the Federal Reserve System (2006)
13. Klee, E.: How people pay: evidence from grocery store data. J. Monetary Econ. **55**(3), 526–541 (2008)
14. Lefebre, P.J.: Digital money – a view from the European Commission. European Business Review **99**(4), 242–256 (1999)
15. McKinsey: European Payment Profit Pool Analysis: Casting Light in Murky Waters, Report (2005)
16. McKinsey: The McKinsey Global Payment Map, Report (2009)
17. Meyer, T.: M-payments: Variety may trump grand schemes, E-Banking Snapshot, Deutsche Bank Research, No 34 (2010)
18. Polasik, M., Maciejewski, K.: Innowacyjne usługi płatnicze w Polsce i na świecie, Materiały i Studia, National Bank of Poland Working Paper Series, No. 241, Warsaw (2009)
19. Polasik, M., Wisniewski, T.P., Lightfoot, G.: Modeling customers' intentions to use contactless cards. Int. J. Bank. Account. Finance **4**(3), 203–231 (2012)
20. Quaden, G.: Costs, advantages and disadvantages of different payment methods. Report, Bank of Belgium (2005)
21. Smart Card Alliance: Contactless Payments: Delivering Merchants and Customer Benefits, A Smart Card Alliance Report, http://www.smartcardalliance.org/resources/lib/contactless_pmt_benefits_report.pdf (2004)
22. Schuh, S., Stavins, J.: Why are (some) consumers (finally) writing fewer checks? The role of payment characteristics. J. Banking & Finance **34**(8), 1745–1758 (2010)
23. Von Kalckreuth, U., Schmidt, T., Stix, H.: Choosing and using payment instruments: evidence from German microdata, European Central Bank Working Paper Series, No. 1144 (2009)
24. Womack, S.: Payment on tap, The Mail on Sunday, 8 August, 79 (2010)
25. Zinman, J.: Debit or credit? J. Bank. Finance **33**, 358–366 (2009)

Part VI

Human-Computer Interaction

User-Centric Data Integration with the MappingAssistant

Heiner Stuckenschmidt[1,2], Jan Noessner[1(✉)], and Faraz Fallahi[3]

[1] School of Business Informatics and Mathematics, University of Mannheim, 68159 Mannheim, Germany
{heiner, jan}@informatik.uni-mannheim.de
[2] Institute for Enterprise Systems (InES), L 15, 1-6, 68131 Mannheim, Germany
[3] Ontoprise GmbH, An der RaumFabrik 33a, 76227 Karlsruhe, Germany

Abstract. Data integration is the problem of transferring complex data from one into another representation in order to support exchange between different systems. From a technical point of view data integration has intensively been studied. However, less attention has been paid to user-centric aspects of data integration. In this work, we present the MappingAssistant which supports the user in finding and validating mapping rules between heterogeneous data sources. Compared to existing approaches we focus on an user-centric approach where the user inspects the consequences of the data integration rules on the instance level rather than being confronted with complex data integration rules. We performed a study which shows that the user-centric approach leads to better integration results, especially for users with little or no technical knowledge and is perceived as being more intuitive.

Keywords: Data integration · User study · Software tools · Usability

1 Introduction

The problem of information integration is omnipresent in information systems and can be seen as one of the major challenges, both, on the technical and the organizational level. In this paper, we focus on the problem of transferring complex data from one into another representation in order to support exchange between different systems, also referred to as data integration.

The problem of data integration has been studied intensively on a technical level in different areas of computer science [1,2]. Researchers have investigated the automatic identification of semantic relations between different datasets [2] as well as the representation and use of identified relations for data transfer and query answering [3]. A prominent line of research investigates the use of ontologies - formal representations of the conceptual structure of an application domain - as a basis for both, identifying and using semantic relations.

In contrast to this work, we are more interested in data integration as a task and in how we can empower the user to solve this task more efficiently and effectively. A successful solution to this problem would have significant implications for data integration in industrial practice. Traditionally, data integration

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 323–339, 2013.
DOI: 10.1007/978-3-642-40654-6_20,

is done by computer science experts of an enterprise or even outsourced to a service provider specialized in solving data integration problems. The fundamental problem of this approach is the fact, that the integration experts are often not experts with respect to the data that is to be integrated. This means that their ability to identify conceptual errors within the integrated data is limited. As a consequence, errors are often found by the user when the data has already been migrated. Fixing such problems at this point typically requires intensive communication between the user and the integration expert and causes overhead. This efficiency loss could be avoided if the user, who knows the data, but not necessarily the underlying technology, would be able to identify and fix integration problems directly.

Following the design science approach [4], we designed and implemented a user-centric data integration tool called MappingAssistant[5,6].[1] This tool allows the user to specify and validate semantic mappings between two datasets following an interactive process model: after specifying a semantic mapping, the system automatically translates data using the specified mapping and presents selected results of this translation to the user, who can then mark individual results as correct or incorrect. Based on this user input, the system identifies mistakes in the semantic mappings by asking the user about the correctness of certain statements dealing with the conceptual model of the data.

In this paper we report the results of a user study that compared the user-centric integration method implemented in the MappingAssistant tool with the use of a pure mapping editor that does not interact with the user. We show that the use of the MappingAssistant approach significantly improves the performance of human users and that the method is especially suited for supporting users with few technical skills.

The paper is organized as follows. In Sect. 2 we discuss the concept of user-centric data integration and briefly review related work on the topic. Section 3 briefly introduces the MappingAssistant tool that we created. The user study evaluating the MappingAssistant tool, which is the main topic of this paper, is described in Sect. 4. We present our research hypotheses, define the experimental setting and the used datasets. Section 5 presents and discusses the results of the study. The paper closes with some conclusions in Sect. 6.

2 User-Centric Data Integration

Studying data integration from a user point of view is a relatively new field of investigation. Traditionally, user studies have rather focused on the problem of personal information management. Data integration can be seen as an important aspect of personal information management [7], however, research has focused on other aspects like the organization of emails or documents. In a recent study, Gass and Maedche have investigated the problem of data integration in the

[1] The interested reader is referred to http://www.ontoprise.de/de/forschung-und-entwicklung/ mappingassistant/ for further information. An illustrative video is available at http://www.youtube.com/watch?v=72abBBTfl_E.

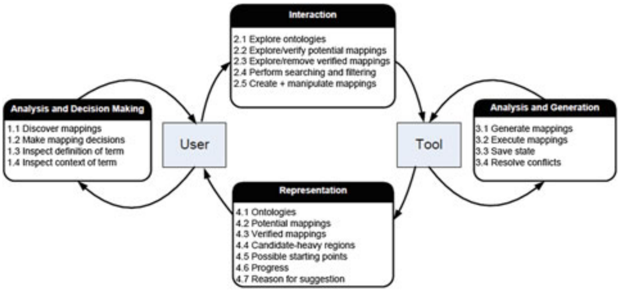


Fig. 1. The cognitive support model for data integration by Falconer [9].

context of personal information management from a user-centric point of view [8]. The scenario addressed in their work, however, focuses on the integration of rather simple data schemas, in that case personal data where the task is mainly to map properties describing a person (e.g. name or bank account number). In many data integration scenarios like product data integration or computer aided design and manufacturing we face much more complex conceptual models and mapping rules.

Recently, researchers in ontology and schema matching have recognized the need for user support in aligning complex conceptual models [10,11]. Most approaches are based on advanced visualization of the models to be integrated and the mappings created by the user [12]. While the appropriate use of visualizations is known to be a key aspect for successful manual data integration, visualizations quickly reach their limits in the presence of very complex or very large models.

As a result, recent work tries to go beyond pure visualization support and tries to include cognitively efficient interaction strategies to support the user [9]. Falconer proposes an interactive strategy for data integration where the integration task is distributed between the user and the tool (compare Fig. 1).

3 The MappingAssistant Tool

In data integration much work has been invested in producing data integration rules with ontology matching systems automatically [13]. However, these rules are still error-prone and, therefore, need to be supervised by a human domain expert. This supervision should be effectively supported by applications.

Existing applications like AgreementMaker [14] present the generated data integration rules directly to the user. Although these tools try to visualize complex data integration rules in an easy understandable way, the evaluation of these rules still requires a significant amount of expert knowledge. Furthermore, in real-world scenarios users are usually confronted with ill-labeled concepts, making the analyses even more complex and time-consuming.

The approach of MappingAssistant [5,6] simplifies the alignment evaluation process by investigating the direct consequences of the data integration rules.

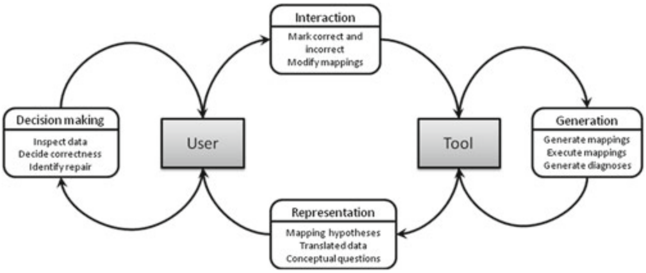


Fig. 2. Modified cognitive support model implemented in the MappingAssistant Tool.

In particular, the MappingAssistant approach allows to evaluate the instance data like product numbers or customers directly, rather than analyzing complex data integration rules. One of the advantages is that in real-world scenarios the domain expert often has sophisticated knowledge about the instance data in his domain.

The MappingAssistant approach implements the cognitive support model for data integration by Falconer (compare Fig. 2). In the *decision making process*, the user inspects the data and decides which concept he wants to examine. In the example shown in Fig. 5 the user selected FamilyCar in the target schema. In the *interaction* step, the user identifies those instances which have been classified incorrectly and marks them as correct or incorrect. Due to the amount of instances a user can be faced to diagnose we utilize different clustering techniques in order to reach data simplification. Attribute-driven combinations of weighted hierarchical and partial clustering algorithms, as mostly described in [15], are therefore utilized. In our example of Fig. 5 the `MX5_Mieta` is a two seated car and, thus, not a `FamilyCar`.

In the *generation* phase a diagnostic algorithm computes the minimal amount of user questions which are needed in order to determine wrong rules or facts. When the user depicts an instance as incorrect, we generate a proof-tree for the corresponding concept-assertion like `FamilyCar(MX5_Mieta)` in our example. Since the user evaluation is correct by assumption, the prolog-based proof-tree must contain at least one wrong node. In order to determine this wrong node our approach traverses the proof-tree in a way that the amount of user questions are minimized for correct as well as for incorrect answers of the user.

In the *representation* phase, the MappingAssistant tool generates questions based on the information it gets from the proof-tree algorithm before. These questions are represented to the user in natural language based sentences in a todo-list, as shown in Fig. 5. If the wrong rule or fact already has been determined the algorithm terminates. Otherwise the user is faced with the next question, which was determined by the diangostic algorithm in the *generation* phase. The approach is implemented as an extension of the OntoStudio Ontology Engineering Workbench [16].

4 A User Study

As part of the MappingAssistant project, we carried out a study in user-centric data integration. The goal of the study was to show that an interactive approach to data integration leads to better results than traditional approaches. In the following we discuss the goals, the design and the results of this study in more details. In Subsect. 4.1 we first define the hypotheses tested in the study in more details, we then present and justify the experimental design in Subsect. 4.2 and present the dataset used in the experiment in Subsect. 4.3. Finally, Subsect. 4.4 provides demographic information about our subjects.

4.1 Hypotheses

The user study was carried out to establish the general hypotheses of our work, which can be phrased as follows:
The cognitive support model helps users to correctly and efficiently integrate complex data.
We have to further substantiate this hypothesis in several ways to arrive at a useful experimental design. In particular, we have to be more explicit about the nature of the support model, the kind of users addressed as well as the integration task to be solved and the notions of efficiency and correctness. In the following, we thus provide more concrete definitions of the hypotheses to be tested.

Cognitive Support Model. We consider the extended cognitive support model in the way it is implemented in the MappingAssistant System (Fig. 2): The system generates mapping hypotheses and executes them. Traditionally, automatic generation of such mappings are generated by lexical and/or tree structure based matching algorithms as described in [2]. The results are represented in terms of translated data instances. The user inspects the translated data and decides on the correctness of data items thereby providing feedback to the system. The user thereby triggers a second interaction cycle, where the system asks questions about the mappings and the underlying conceptual model waiting for the user to answer them.

Propective Users. The motivation for designing the extended cognitive support model and for implementing the MappingAssistant was to enable users with little or no technical knowledge in data integration. Thus, our refined hypothesis is that users with limited knowledge in conceptual modeling and data integration show a better performance when supported by the cognitive support model. Further, we assume that the positive effect is stronger for people with very little knowledge than it is for users with more knowledge in these areas.

Integration Task. We decided to focus on the task of validating an existing set of mappings rather than creating a new set of mappings. If each subject is asked to *create a new* set of mapping, we would get many different solutions which might all be correct. Especially when complex data structures are involved the same integration task can be solved using different sets of mapping rules. This makes it extremely difficult to measure the correctness and completeness of the solution provided by a user. In contrast to this, identifying errors in an *existing* set of mapping rules is a more well-defined task that has a unique solution, provided the test data is designed in a suitable way.

Quality Criteria. We expect two positive effects of using the cognitive support model. The first one is efficiency, which means that a user is able to find errors in a set of mapping rules in a shorter period of time. The second one is effectiveness: we assume that a user is able to find more errors with the MappingAssistant approach, which would have remained undiscovered without the support by the system.

In the context of a controlled experiment, it is hard to distinguish these two effects; our hypothesis is that users are able to find more errors in a fixed period of time.

4.2 Experimental Design

In the following, we describe the experimental design of a user study we carried out to test our hypotheses.

Study Design. The general idea of the experiment is to compare the MappingAssistant approach to data integration with a conventional approach that is solely based on the use of a mapping editor. As a result, the study consists of two tasks, both concerned with identifying errors in a set of mapping rules that combines two conceptual schemas. In order to control external parameters of different integration tools, both tasks are carried out using the OntoStudio knowledge Engineering environment. OntoStudio contains a mapping editor that can be used to visually inspect and modify a set of mapping rules as well as an extension that implements the MappingAssistant approach.

Based on this technology we created the study design depicted in Fig. 3.

Participants have to solve two integration tasks in a timeframe of 10 min each. Before each task, the participants are briefed about the task. The information given to the user consists of an information sheet explaining the task and instructions by a supervisor who answers questions about the task without providing hints towards the expected results. The timeframe for this instruction phase has been determined individually for the two tasks, but is the same for all participants.

The order in which the two tasks are carried out is switched after every subject in order to avoid an efficiency bias for the second task due to a training effect. Since all participants perform both tasks, we do not need to divide the

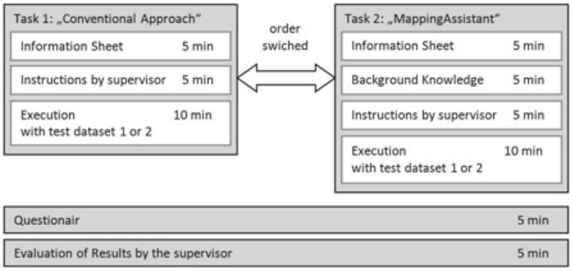


Fig. 3. Design of the user study.

users in groups, but can compare the performance and experience of the users directly.

Furthermore, two different datasets are used, which are assigned to the two different tasks randomly for the purpose of ruling out a possible bias due to a different level of difficulty. After a subject has carried out both tasks, he or she is asked to fill in a questionnaire on the perceived difficulty and support by the tool as well as on the level of expertise of the subject.

Thus, in our study the *independent variable* is determined by either using the conventional approach (Integration Task 1) or the MappingAssistant approach (Integration Task 2). The *dependent variable* is the number of errors the subjects found in the respective dataset [17].

The following subsections provide more information about the two integration tasks, data and subjects used in the study as well as on the contents of the questionnaire.

Integration Task 1 (Conventional Approach). Integration Task 1 consists of using the conventional user interface of the OntoStudio Mapping Editor to discover errors in a set of mapping rules. Figure 4 shows the user interface used in this task. It shows the conceptual model of the source data on the left and the source model on the right hand side. The models consist of classes, relations and instances.

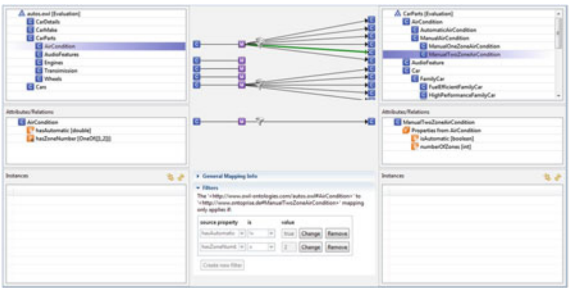


Fig. 4. Traditional user interface for creating and evaluating semantic mappings.

The mapping rules connecting the two models are displayed visually in the middle of the screen. Below the visual representation of the mappings, specific filter conditions for mapping rules can be displayed in the form of pre-selected drop-down menus by clicking on a mapping rule.

The task of the user is to inspect the mapping rules for errors by navigating the conceptual models and the mapping rules and their filter conditions. This standard configuration of the tool does not provide special functionality for focusing on problematic issues, meaning that the user has to actively search for errors without being guided by the tool.

Integration Task 2 (MappingAssistant Approach). The second task consists of solving the same problem, namely the identification of errors in a set of mapping rules. However, instead of the conventional mapping editor, the MappingAssistant plug-in is used. Its interface also shows the conceptual model of the source and the target data set and the mappings between the elements. Instead of the filter conditions of the rules, however, the interface shows results of translating data using the mapping rules, as well as a todo-list with questions generated by the tool that have to be answered by the user (compare Fig. 5). Additionally, the plug-in allows for utilizing different clustering techniques in order to reach data simplification on the instance level.

The task of the user is again to inspect the mappings for errors. This time, however, an interactive process is used. The user actively selects a class in the target model and inspects the instances that have been created by executing the mapping rules. Based on his or her knowledge of the domain, the user can mark rows in the data table as incorrect indicating that they are not mapped correctly to the selected class in the target model. His action then triggers a diagnosis procedure that generates yes/no questions about the conceptual structure of the data and displays the questions in the todo-list. The user has to answer these questions thereby guiding the semi-automatic diagnosis process to the errors.

Fig. 5. Interactive user interface of the MappingAssistant data integration tool.

Once an error has been found, the user can select another class in the target schema and so forth.

Questionaire. The questionnaire that had to be filled in by all participants consisted of four parts. All questions except demographic questions had to be answered on a 1–5 scale:

1 definitely disagree,
2 rather disagree,
3 neither agree nor disagree,
4 rather agree,
5 definitely agree.

In the following, the different question categories are discussed:

Previous Knowledge. This category contained questions about the knowledge and expertise of the subject in the areas of data modeling and data integration. Examples of questions from this area were:

– I am experienced with using complex software tools
– I am used to apply filter rules for selecting data (e.g. in Microsoft Excel)
– I have good knowledge about formal data models
– I have good knowledge in data matching and integration

Task 1. This part of the questionnaire explicitly addresses the experiences of the subjects with respect to performing Task 1. The goal was to get a better idea of the perceived complexity and difficulty of this task. Examples of questions were:

– I was confused by the representation of the mapping rules
– I was able to decide on the basis of the filter expressions whether a mapping is correct or not
– It was easy to work off the mappings without missing out on something

Task 2. This part focused on the experience of the subjects with solving task 2 and being supported by the MappingAssistant tool. The goal was to judge the level of support provided by the system. Example questions were:

– It was easy to identify wrong instance data
– The presentation of wrong mapping rules by the system was intuitive
– The attributes of translated data items helped me to identify mistakes

Comparison of Task 1 and Task 2. In order to be able to compare task 1 and task 2 the following question was asked in both tasks:

– The handling was intuitive

Demographics. Finally some demographic information was asked including age, gender and occupation of the subjects.

4.3 Datasets

When selecting the datasets for the study, we had to find a trade-off between the following issues. On the one hand having a problem that can be understood and solved within the limited timeframe of the study and on the other having enough complexity to adequately represent a realistic data integration challenge and supporting our assumption that our method works well for complex problems. In order to be able to satisfy these needs we decided to use a combination of existing data and data that has been created manually for the study.

Source Dataset. We decided to use a technical domain because of the typical rich feature-sets and complex definitions. As it turned out that real world datasets were much too large and complex to be handled in the study, we chose an instructional dataset from the car-selling domain[2] that was automatically converted to fit the data model of the OntoStudio Tool. The dataset contains 324 data records that are described using more than 100 attributes. In addition the data is organized in a concept hierarchy containing 91 concepts. This makes the dataset complex enough to pose a real data integration challenge, but as we could also confirm in the study, small enough to be handled in a limited amount of time.

Target Schema. In order to be able to control the experiment and include different types of errors in the alignment, we manually built a target schema to which the records of the source dataset need to be translated. Building the schema, we followed established best practices for conceptual modeling.

The resulting schema consists of a 29 classes organized in a hierarchy (compare Fig. 6) and 20 attributes for describing data records.

Mapping Rules. We manually created a set of mapping rules between the two schemas and validated the correctness of the mappings with respect to correctly translated data. Based on this correct mapping set, we created two mapping sets each of which contain ten errors. We introduced errors with different level of complexity. The main task of the participants in the study was detecting as much of the produced errors as possible within the limited timeframe.

The simplest type of wrong mapping rules are rules connecting classes that are not identical. An example is the mapping rule `wheel` on `engine`. We assume that these kinds of errors are easy to spot even by inexperienced users. Out of ten errors each dataset contained four errors of such a kind.

The second type of errors was introduced by modifying the filter conditions associated with a correct mapping rule. An example would be the rule mapping `AirCondition` to `AutomaticOneZoneAirCondition`. The correct filter conditions for this mapping rule are `hasZoneNumber = 1` and `hasAutomatic = true`. In this case we modified the filter conditions to `hasZoneNumber = 2` and

[2] http://gaia.isti.cnr.it/∼straccia/download/teaching/SI/2006/Autos.owl

Fig. 6. Class hierarchy of the manually created target schema.

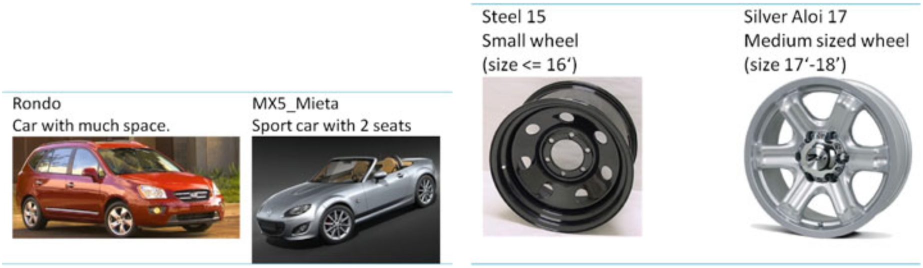
`hasAutomatic = false`. We assume that these kinds of errors are harder to identify by the user because it requires a detailed investigation of the mapping definition. Each of the two mapping sets contained six out of ten errors of this type.

Domain Information. Participants were provided with background knowledge about the domain of interest. The knowledge consisted of information about specific car types and car equipment. The respective information was provided in terms of images in combination with short descriptions. Figure 7 show examples of such information for cars and wheels.

4.4 Participants

Twenty-two subjects participated in the user study. Six of the subjects were female and sixteen male. The average age of the subject was 27.8 year with the youngest subject being 21 and the oldest over 50. About half of the subjects (54 % of the persons involved in the study) were students.

All participants were used to utilize complex software tools (average: 4.91). On average subjects had medium experience with using filter rules (average: 3.72), conceptual models (average: 3.13), and data integration (average: 2.72) in the past.



(a) About Cars (b) About Wheels

Fig. 7. Simulated background knowledge.

In all three cases, the answers ranged from 1 to 5 providing a good coverage of different levels of expertise. In particular, the variance of the subjects' experience using filter rules was 1.92, using conceptual models was 2.22, and using data integration was 1.79.

5 Experimental Results

We analyzed the results of the study with respect to the quality of the results produced by the participants, the correlation of the results with the level of expertise and the perceived support by the system. Since all users perform both integration tasks, we can directly compare the users' performance on the respective integration tasks.

The detailed results are presented in the following subsections.

5.1 Quality of the Results

We measure the quality of the results for an individual participant by comparing the errors identified by the user with the set of errors previously introduced in the mapping set. We use well known quality measures from the area of information retrieval, more specifically precision, recall and F^2-measure. Before presenting the results, we briefly recall the definition of these measures.

Let TP denote the number of true positives, that is the number of errors that have correctly been identified by a subject, FP the number of false positives that is the number of mapping rules that have falsely been identified by the subject as an error and FN the number of false negatives, namely the number of errors that have not been found by the subject. Following this definition precision (P), recall (R) and F^2 measure (F^2) are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

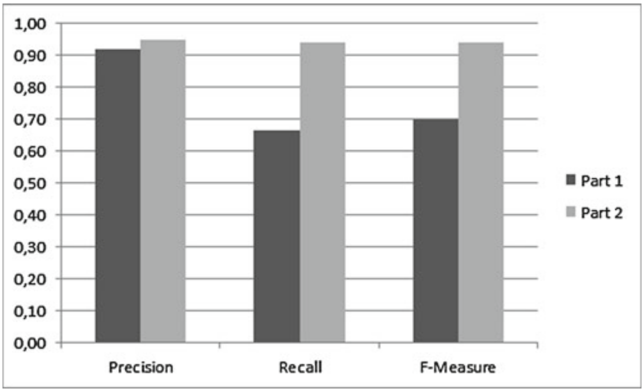


Fig. 8. Average performance of subjects on the two integration tasks.

$$F^2 = \frac{(1+2^2)P \cdot R}{(2^2)P + R} = \frac{5P \cdot R}{4P + R} \tag{3}$$

Figure 8 compares precision, recall and F2 measure that have been achieved by the subjects on average for the two integration tasks.

The results show that there was no significant difference between the conventional tool and the MappingAssistant approach with respect to precision. Both values are close to 1.0 meaning that subjects almost never identify mappings as errors that are actually correct. We can see however, that there is a significant difference with respect to recall. Using conventional technology, the subjects were only able to find two thirds of the errors on average. In comparison to that, using our approach the yielded recall was much higher than utilizing conventional technology. In particular, the subjects on average were able to find nine out of ten errors within the given time frame of 10 min.

The advantage of the MappingAssistant approach with respect to identifying existing errors more efficiently becomes even more obvious when looking at Fig. 9.

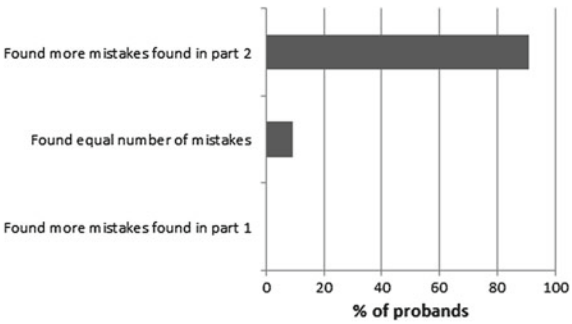


Fig. 9. Comparison of performance on subject level.

It shows that 91 % of the subjects found more errors using MappingAssistant than with the conventional tool. 9 % of subjects found the same number of errors and none of the subjects showed a better performance with the conventional technology.

The results clearly show that our method helps users to find more errors in a given period of time thereby confirming our hypothesis that the method increases efficiency and effectiveness of data integration.

5.2 Correlation with Level of Expertise

In order to test our hypothesis that our method in particularly supports users with limited technical knowledge, we compared the performance of participants with their previous knowledge in conceptual modeling and data integration. When comparing previous knowledge in conceptual modeling and data integration with the overall performance in task 1, we can see that there is indeed a relation between these two aspects (compare Fig. 10).

We can conclude that for successfully performing data integration with a conventional tool a high degree of expertise in conceptual modeling and data integration is needed. While the trend is not that clear with respect to previous knowledge in conceptual modeling, the result is more conclusive with respect to previous knowledge in data integration.

Figure 11 shows the relation between the level of previous knowledge and the performance improvement achieved by using the MappingAssistant approach.

With the MappingAssistant approach the subjects achieved high F^2 measure results. The results of the test persons were independent from the previous degree of expertise in conceptual modeling as well as data integration.

Although the results are not as clear as for the quality improvement, we can observe that the performance using the MappingAssistant is independent from previous knowledge. In summary, these results confirm our hypothesis that a user-centric interactive approach to data integration has a stronger positive effect for people with little technical knowledge compared to the conventional approach.

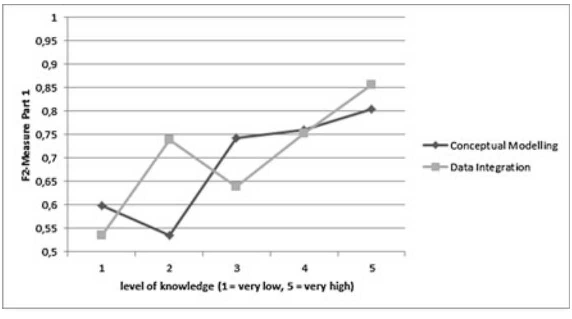


Fig. 10. Relation between previous knowledge and performance in integration task 1.

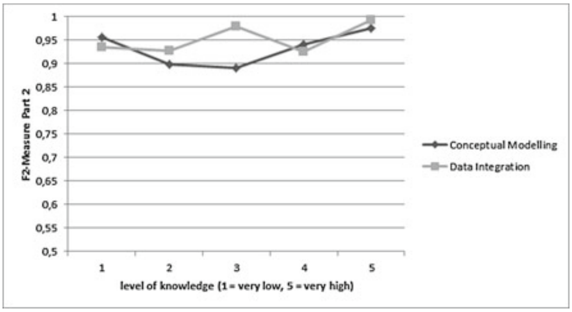
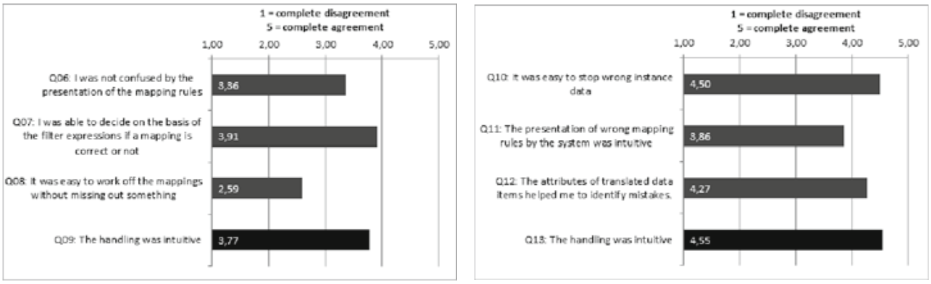


Fig. 11. Relation between previous knowledge and performance in integration task 2.



(a) Part 1 (b) Part2

Fig. 12. User feedback.

5.3 Qualitative Results

In addition to the objective results, we were also interested in how the different tasks were perceived by the users. In particular, we wanted to find out, how intuitive the user interface was and how the users judged the difficulty and the support by the systems. In the following we present the answers to several questions related to the mentioned aspects.

Figure 12 shows the average answer of the users with respect to task 1 and task 2 according to the 1–5 scale where 1 means *complete disagreement* and 5 *complete agreement*.

Concerning task 1 the results show that users found difficult to complete the task without missing mapping rules. With respect to the other questions about the difficulty of the task and the design of the user interface, the users were rather indifferent. In the results of task 2 we see that in general there is a much higher agreement of users with questions concerning the benefits of the approach. In particular, users considered the MappingAssistant approach to be more intuitive than the traditional one (3.77 vs. 4.55 average score) supporting our hypothesis.

While the results are not directly comparable, we can still conclude that our assumption about the benefits of the MappingAssistant approach being more intuitive and easier to follow by the users is shared by the subjects of our study.

6 Conclusions

Data integration is a difficult task that typically requires substantial knowledge not only of the data to be integrated but also of data integration technologies. The goal of our research was to enable the people with less or no knowledge of these technologies to integrate their data. We presented a user-centric approach to data integration that is based on a cognitive support model, which has been implemented in the MappingAssistant data integration tool. We presented the results of a user study demonstrating that the approach empowers users to solve data integration problems more effectively and efficiently. In particular, we showed that users were able to find more errors in mapping rules in a given period of time. Further, we were able to show that while with conventional mapping technology a high level of expertise in mapping technology is required, the MappingAssistant approach significantly reduces the performance difference of experienced and inexperienced users. Finally, we could show that users considered our approach to be more intuitive.

We believe that the user-centric approach to data integration presented in this paper can have a real impact on the practice of data integration in enterprises. In particular, the approach can help expert users within an organization to retain more responsibility for data integration. While today, data integration tasks often either have to be delegated to the IT department or even be outsourced to specialized companies, our technology can enable users to perform non critical data integration tasks themselves. This can save time and money in enterprise data integration. Furthermore, it can create more options for on the fly data integration or mesh-ups that can provide useful information but are not needed on a regular basis.

In future work we will extend the presented diagnosis component of the MappingAssistant with an induction component. This induction component will provide suggestions for repairing the wrong mapping rule which has previously been found with the diagnosis component.

References

1. Halevy, A., Rajaraman, A., Ordille, J.: Data integration: The teenage years. In: Proceedings of the 32nd International Conference on Very Large Data Bases, VLDB Endowment, pp. 9–16 (2006)
2. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, New York (2007)
3. Bellahsene, Z., Bonifati, A., Rahm, E.: Schema matching and mapping. Int. J. Comput. Appl. **34**(8), 14–21 (2011)
4. von Alan, R., March, S., Park, J., Ram, S.: Design science in information systems research. Mis Q. **28**, 75–105 (2004)

5. Noessner, J., Fallahi, F., Kiss, E., Stuckenschmidt, H.: Interactive data integration with mappingassistant. Demo Paper at the 10th International Semantic Web Conference ISWC (2011)
6. Fallahi, F., Noessner, J., Kiss, E., Stuckenschmidt, H.: Mappingassistant: Interactive conflict-resolution for data integration. Poster at the 8th Extended Semantic Web Conference, ESWC (2011)
7. Teevan, J., Jones, W., Gwizdka, J.: Personal information management. Commun. ACM **49**, 68–73 (2006)
8. Gass, O., Maedche, A.: Enabling end-user-driven data interoperability - a design science research project. In: Proceedings of the 17th Americas Conference on Information Systems (AMCIS 2011), Detroit, USA (2011)
9. Falconer, S., Noy, N.: Interactive techniques to support ontology matching. Schema Matching Mapp. **9**, 29–51 (2011)
10. Falconer, S.: Cognitive support for semi-automatic ontology mapping. Ph.D. thesis, University of Victoria (2009)
11. Falconer, S.M., Storey, M.A.: A cognitive support framework for ontology mapping. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 114–127. Springer, Heidelberg (2007)
12. Granitzer, M., Sabol, V., Onn, K.W., Lukose, D., Tochtermann, K.: Ontology alignment - a survey with focus on visually supported semi-automatic techniques. Future Internet **2**, 238–258 (2010)
13. Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., Meilicke, C., Nikolov, A., Pane, J., Sabou, M., et al.: Results of the ontology alignment evaluation initiative 2010 (2010)
14. Cruz, I., Antonelli, F., Stroe, C.: Agreementmaker: efficient matching for large real-world schemas and ontologies. Proc. VLDB Endowment **2**(2), 1586–1589 (2009)
15. Hair Jr., J., Anderson, R., Tatham, R., Black, W.: Multivariate data analysis (1995)
16. Maier, A., Schnurr, H.-P., Sure, Y., et al.: Ontology-based information integration in the automotive industry. In: Fensel, D., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 897–912. Springer, Heidelberg (2003)
17. Wohlin, C.: Experimentation in software engineering: an introduction. Springer, Heidelberg (2000)

Introducing New Technology in Educational Contexts: Schools as Organizations

Elaine C. S. Hayashi[1(✉)], M. Cecília Martins[2], and M. Cecília C. Baranauskas[1,2]

[1]Institute of Computing, University of Campinas, Campinas, Brazil
{hayashi, cecilia}@ic.unicamp.br
[2]NIED, University of Campinas, Campinas, Brazil
cmartins@unicamp.br

Abstract. The school context offers many opportunities to explore the use of digital technologies. When designing for schools, systems designers usually tend to consider only the students or, at most, teachers and students. However, schools are complex social organizations that are composed by many other interested parties. When aiming at solutions that make sense to the users, the entire school, as an organization, should be considered. This paper, presents the activities we conducted at an elementary public school in Brazil, facing the challenge of constructing meaning for a new digital artifact. For the analysis of organization and context we used adapted practices from Organizational Semiotics (OS) and Participatory Design (PD). Our results suggest that both OS and PD are suitable for educational projects and they can anticipate problems and propose solutions for prospective use of technology in such contexts.

Keywords: Human-computer interaction · Participatory design · Organizational semiotics · Educational technology

1 Introduction

Technology is everywhere in our lives, mediating our actions. Hence, the use of technology should be a powerful tool to provide access and to promote the construction of knowledge in the schools. This is especially true in the context of developing countries, where the opportunities of access to digital technology at home are scarce. However, to be incorporated into schools' practices, it needs to make sense to the community of users. Technology use should be transparent, providing teachers and students with learning opportunities, so that a digital culture might be created at school and perhaps disseminated to the schools' physical surroundings.

According to a survey conducted by the Brazilian's National Institute for Educational Studies Anísio Teixeira – INEP, and the Brazilian Ministry of Education and Culture—MEC [20] roughly 31,700,000 students are enrolled in the fundamental level, from which around 27,500,000 are in public schools. A survey conducted by the Brazilian Internet Steering Committee - CGI [6] indicated that while 81 % of

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 340–357, 2013.
DOI: 10.1007/978-3-642-40654-6_21,

Brazilian public schools (including fundamental level and high schools) have a computer lab, only 4 % of the public schools have a computer in the classroom.

Recently, a program from the national government has proposed the use of educational laptops in a 1:1 model and has been incrementally distributing the machines to public schools. The mobility of the laptop allows students to take it to their homes, extending the potentials of the laptop to the students' family. In a country where 55 % of the population in general have never used a computer [6], that initiative might represent a relevant step towards more access to information and knowledge, i.e., a fairer society.

For a new digital technology to be effectively incorporated in educational contexts it must make sense to all involved parties and it must consider their habits, abilities and organizational culture. On that ground, we believe that the process of bringing this new artifact of technology to the school context must happen under a socio-technical approach. This situation presents the research challenge of formalizing models and techniques to promote understanding of the situated scenario towards meaningful use of technology in schools.

1.1 Context

In January 2009, 500 XO laptops were donated by the One Laptop Per Child [25] organization for a research project at a school in the suburban area of the city of Campinas, in São Paulo. This project [33] is a research effort that runs in parallel and independently to other government's initiatives. In the approach adopted by the government, to insert the laptops at schools, a same methodology is applied to all the schools from different regions of the country. Differently, our approach acknowledges the situated character of the problem and constructs a methodology based on a joint effort of the different parties involved: researchers, designers, developers, educators, school staff, and students.

Taken the school as a complex organization, the frame of reference to our work is based on methods and artifacts of the Organizational Semiotics (OS) [30, 31], combined with techniques inspired by Participatory Design (PD)[23]. Both OS and PD are articulated to compose the collaborative practices of Semio-participatory Workshops (SpW) [3] conducted within the school.

This paper presents our findings from the initial stages of the process of clarifying the problem of technology embedding in a fundamental public school in Brazil, on the grounds of OS. The paper captures the impacts of the SpW based methodology and discusses results of the first out of ten other workshops that took place between years 2010 and 2012.

These workshops are part of the 'XO Project' [33], which promotes reflection about educational laptops in the Brazilian context, investigating the use of a new technology within a school's daily reality. Results from the XO Project are expected to contribute to the development of the triad technology, education and society. More specifically, the XO project aims at proposing a model for the use of educational laptops at Brazilian public schools based on a collective construction.

The XO Project connects researchers from different areas (e.g. Computer Science, Pedagogy, Psychology, Medias) and institutes (NIED – Information in Education Nucleus – and the Institute of Computing, both from the University of Campinas, in Brazil). The school (EMEF Padre Emílio Miotti) is a public school located in a low income neighborhood. This school was selected with the assistance of the Municipal Secretariat of Education.

The paper is organized as follows: related approaches are reviewed in Sect. 2. Theoretical and methodological framework adopted by the project is detailed in Sect. 3. Section 4 describes the planning of the workshop. The workshop itself, with its results and discussion, is detailed in Sect. 5. Section 6 reports the experience gathered in the two years that followed the first workshop. Section 7 concludes.

2 Related Work

Since the proposal of the XO laptop by the OLPC in 2005, many initiatives have taken place to investigate its use at schools. Not only the XO laptop is being target of studies, but also similar technology that has been proposed after OLPC.

From a pure technical perspective, Moody and Schmidt [22] present the advantages of wireless networks in education and numerate some concerns to be addressed before the wireless networks are implemented at schools.

From a socio-technical perspective, Cervantes et al. [5] analyzed the social and technical infrastructures that support the use of low cost laptops at schools. That was done by observing the activities that took place at elementary schools in Mexico after the laptops were already distributed. The authors described, based on what they saw in the schools, how the available infrastructures (both from technical and human perspectives) shaped the use of the laptops.

Also describing the laptops use after its implementation, Flores and Hourcade [9] report on the experience in Uruguay. The government in Uruguay has distributed laptops to every child in the public elementary school of grades 1–6 in the country. The authors describe the first year with the laptops at the schools.

In the Brazilian context, Corrêa et al. [7] conducted surveys similar to market researches. Qualitative (interviews) and quantitative (forms) approaches supported the investigation on the acceptability of low cost laptops among teachers and students before introducing the laptops to the interviewees. This descriptive study collected and reported on teachers' and students' beliefs on the impacts of digital technology at school.

The important difference between the approach of Corrêa et al. and ours lays on the role that teachers and students play in the project. Instead of passive informants, we take teachers, students and other members of the social organization formed by the school, as active partners. More than eliciting participants' concerns, our objective is to promote a collective awareness, encouraging a collaborative prospection of ideas and solutions, creating together more meaningful uses of digital technology, even before the arrival of that technology.

Expanding from the situated context of educational laptops to the general use of Information and Communication Technologies (ICT) in educational settings, Lim [14] also argued for the importance of taking socio-technical perspectives. Based on the Activity Theory, the author proposes a theoretical framework that shows the connection of ICT with learning and the sociocultural setting. The garden-as-culture metaphor from Cole (apud [14]) is adopted to provide a broader view of the school and educational system in the society at large.

Although OS is not mentioned, the author's [14] figure displays some ideas that are similar to those present on OS artifacts: in the framework from Lim, the garden metaphor with the activity system is shown as embedded circles, with the society as the outermost circle. Those more formal structures are on inner circles; and the activity system itself as the innermost circle.

Explicit reference to OS is made in the work of Melo et al. [19]. Different design techniques and artifacts are combined into the model that the authors propose for the design centered in children's participation. The work of Melo et al. [19] had its focus on the process of design for and with children towards interfaces that made more sense to the children.

Our work faces the challenge of providing more meaningful appropriation of technology within a school community. The simple injection of a foreign technical device into a community's life seems easier but the adjustments demanded by this approach might feel less natural. Our pursuit aims at promoting a collective construction of meanings towards more natural housing of the new technological tool, having the entire community – with teachers, students and other members – as actors.

In this endeavor, we adopted a theoretical and methodological reference based on Organizational Semiotics (OS) and Participatory Design (PD). Even though both OS and PD have origins in industrial and business areas, we argue that they can be successfully applied in educational contexts, guiding the process of technology assimilation to more meaningful results. The principles and some instruments from OS and PD applied in this project are detailed in the next section.

3 Methodology

3.1 Organizational Semiotics

OS views information systems as organizations, composed of socially established models of behavior, beliefs, perceptions and values [32]. In this approach, the design of technology starts with the understanding about the sense that the community of users make of signs and how the organization is structured.

According to Stamper et al. [32], any organization can be described in terms of the norms that govern the behavior of that social group. The authors suggest that such norms are applicable to different types of taxonomies. One possible categorization is by the level of formality of the norms. In this case, the categories are: technical, formal and informal. Strictly precise norms that can be expressed as instructions to be followed by, for instance, a computer, comprise the technical norms. The written norms (i.e., bureaucracy) are the formal ones; and all other norms that people know and live according to are the informal. These levels can be represented as the layers of an onion, where the technical systems are embedded inside the formal and informal organization. The Semiotic Onion [31] comprises the technical, formal and informal layers of real information systems.

Another possible way proposed by Stamper et al. [32] to classify norms is according to their role in relation to signs and their functions, which can be organized using the Semiotic Framework [30]. The authors indicate that this taxonomy helps understanding the impact of information technology when that technology is the cause of organizational change. The Semiotic Framework from Stamper organizes the properties of the signs into six levels (three more levels than the usual semiotic division of syntax, semantics and pragmatics):

- Social level: for a sign to be fully understood, as argued by Stamper [30], one needs to understand its potential or actual social impacts. That includes concerns about ways of behaving, sets of values, shared models of reality, etc.
- Pragmatic level: for a sign to have a use, it must have an intention, shared by its creator and its interpreters. This level involves the understanding of context and forms of communication.
- Semantic level: this level is related to the meanings that are continuously constructed and reconstructed while people use syntactic structures to organize their actions.
- Syntactic level: concerns formal structures that maps or transforms symbolic forms. These are mechanical transformations and they are proper of software developers.
- Empiric level: this level includes the aspects related to telecommunication in general: noise, patterns, redundancy, errors, channel capacity, etc.
- Physical level: the physical properties of objects or events: equipment, hardware, physical structures, etc.

Together, the layers of the Semiotic Framework (also Semiotic Ladder) guide the understanding of how an organization works. Moreover, it helps in analyzing factors that might contribute to more successful processes [16, 30].

In the next sections we exemplify the use of the Semiotic Onion and the Semiotic Framework for the analysis of data collected from the activities in the school. Next we briefly describe two other artefacts that were explored collectively during the encounters with the community of users.

Stakeholder Chart. The various methods that compose MEASUR (Methods for Eliciting, Analysing and Specifying Users' Requirements), proposed by Stamper [30], provide tools for better understanding organizations. Liu explains that, even when dealing with rather chaotic problem situations, the methods allow gradual and precise clarification, until a set of technical solutions can be reached.

The Stakeholder Analysis Chart (SC) is one of the Problem Articulation Method techniques, from MEASUR [17, 30]. The actions of stakeholders, with their roles, interests and responsibilities, usually impact the result of a project. Because of that, it is important to clearly identify who the stakeholders are so that they are properly taken into account in the process.

Evaluation Framing (EF). The EF [2], differently from the Valuation Framing [17], is a technique that aids in the process of anticipating problems new technology might bring to the organization, and prospecting solutions for them. The EF guides a reflection on issues and possible ideas and solutions related to each category of stakeholders raised through the SC. In the workshops that we describe in this paper we have used the EF to help the whole group in the identification of the main issues related to this specific context of new technology and educational change.

3.2 Participatory Design

The Participatory Design (PD) has its origins in Scandinavia in the late '70 s, appearing in the workplace to promote more democratic insertion of technology among those more affected by it [28]. Brought to the design context, the involvement of the users in the design process contributes to the motivation of participants and to greater satisfaction with the outcome, since all are co-authors of the resulting system.

Several techniques have been proposed to allow this interaction between designers and end users, and to allow the participation of users in all stages of the design process. Among the techniques used in PD, we can mention games, plays and different levels of prototyping. Muller brought together and listed several of these activities in [23].

3.3 Semio-Participatory Workshops

The insertion of digital artifacts in the educational environment demands a vision of its socio-cultural context [14].

The articulation of methods from OS with principles and practices from DP represents a powerful tool for the process of understanding the social context while involving the target community in the course of actions. This articulation was

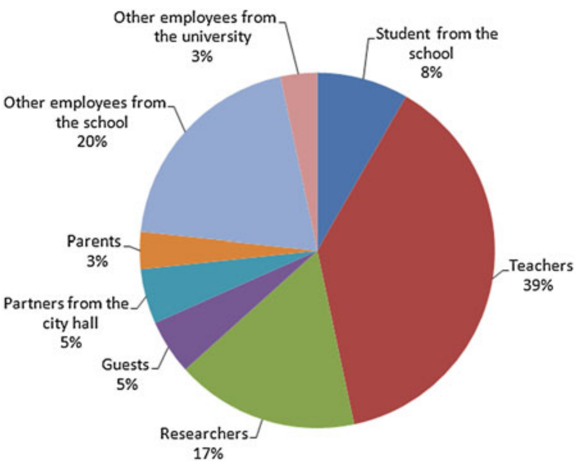


Fig. 1. Participants of the SpW and their roles.

materialized in activities that were carefully tailored for each community who joins the SpW [3].

OS was proposed in the context of information systems and business organizations. PD has also its origins in the work field contexts. However, the combination of OS and PD has shown important results in other knowledge domains and practice areas. For example, Neris el al. [24] report their actions during the design of an inclusive social network, involving a community of young adults and seniors from a low-income neighborhood. Other successful examples include practices related to the domains of critical systems [11], geographic information systems [8, 27], iDTV [10], among others.

Those results encourage us to use the presented methodological frame of reference to bring awareness and collective discussion within an educational organization, involving school-age children and their fellows (teachers, parents and other members of the school). Together, we have been engaged in the process of constructing a more meaningful use of technology. The next section describes how these methods and artifacts have been collaboratively articulated in our scenario.

4 Preliminary Practice

The first encounter with the school's community took place even before the arrival of the laptops at the school. Researchers from different areas (computer science, pedagogy, multimedia, psychology, etc.) gathered with the members of the school, which included teachers, students, parents and other employees from the school (e.g., principal, pedagogue, cook, janitor). Also representatives from the secretary of education of the city hall were present. This first SpW had 60 participants and the composition of this group is represented in Fig. 1.

The discussion on the new technology that was about to be used at the school would be richer if the participants could have a better idea about that technology. Because of that, the SpW started with what was called the "XO Mini Fair".

The mini fair allowed participants to have their first contact with the XO laptop. For this event, eight XO laptops were distributed among four stands. In the first stand, participants had the chance to manipulate the laptop, finding out how to open it, how to close it, use the antennas, rotate the screen.

At another stand, the features of XO's webcam were explored in activities where participants were able to take pictures and make short videos. The other two stands examined the chat activity and the educational game *SOO Brasileiro* [29]. Figure 2a illustrates a moment of this activity in one of the stands. Each stand had at least one facilitator, i.e., one of the researchers who would be available to assist participants in their interactions with the XO whenever needed.

After the XO Mini Fair, the whole group gathered together again and videos were presented. The first video was composed of extracts that formed a shorter version from a video available on Youtube (http://www.youtube.com/watch?v=ZwQOibphtjc). This video showed the experience that a public school in another State in Brazil was having with the use of XO laptops. The second video presented some of the initiatives from our research group related to low cost laptops and the main features of the XO laptops.

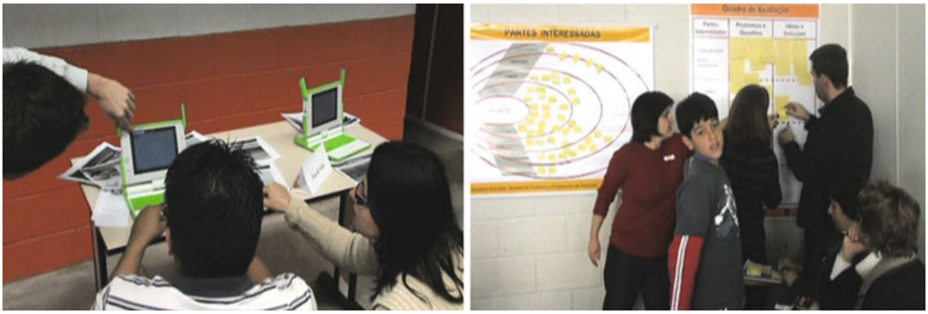


Fig. 2. On the left, (a) participants of the SpW examining the XO at the mini fair; and (b) one of the groups constructing the Evaluation Framing with post-it's.

After all participants were familiar with the laptop and some of its possible uses, they were invited to discuss, in smaller groups, about the impacts of bringing that technology to the school.

Posters had been previously prepared, depicting the empty artifacts of SC and EF. Due to the size of the group (60 people), they were distributed in three smaller groups. Each group had one poster of each artifact; and participants expressed their thoughts writing in post-it's that would be fixed on the posters. In each group, a facilitator led the activity, eliciting responses and attaching the post-its on the charts. Figure 2b illustrates a moment of this activity.

After all charts were created, the entire group got together again and the results were discussed; each group summarized their results on the activities.

Towards the end of the SpW, participants were invited to take a moment of introspection. They were asked to write, individually and anonymously, adjectives or complete sentences that reflected their perceptions and expectation about the project. The activity was not mandatory, but the majority reported on their impressions.

5 Results and Discussion

The XO Mini Fair was an attempt to diminish participant's anxieties about the technology they were going to receive in the school. Participants visited all stands and learned how to manipulate the laptop and how to use some of its main features.

After the SpW, researchers analyzed the material produced during the workshop. The tables that were filled separately by all three groups were combined into one consolidated table.

The charts displayed participants concerns related to varied subjects: safety issues (e.g., precaution and protection of children carrying a laptop from home to school and vice versa); training (e.g., how to train teachers, students, parents and other users); operational issues (e.g., how and where to store all machines, how to distribute them among the children, maintenance issues); among others. Figure 3 illustrates the main issues discussed in the groups, represented and sorted according to the levels of the Semiotic Framework.

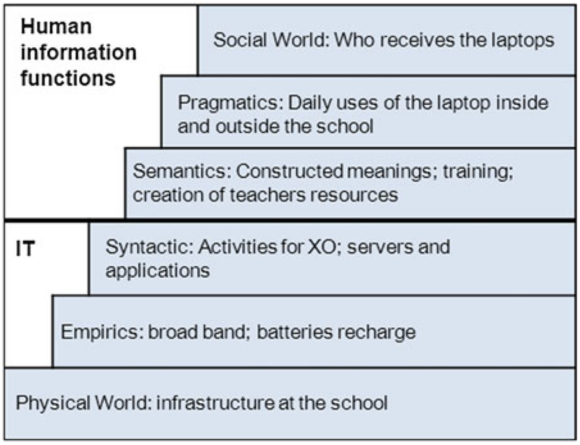


Fig. 3. The Semiotic Framework [15] organizing the main ideas related to the project.

By observing the summarized general results of the SpW in Fig. 3, we can realize how the use of SC and EF helped the group to become aware of issues ranging from the physical to the social domains. Technology related projects are usually directed mainly towards the IT levels (physical, empirics and syntactic levels of the Semiotic Framework). In the context of our project, the concerns with human information functions (the social, pragmatic and semantic levels) are of vital importance. For the technology deployed to make sense, and thus be incorporated into community practices, it is necessary to address issues of higher levels of the framework as well as the issues related to the lower levels. For example, since the number of students in the school exceeds the number of machines, deciding how to distribute the laptops will have an impact in the schools' culture and habits. The infrastructure of the schools (e.g., number of sockets at a classroom, electric capacities, etc.) may influence the daily uses of laptops. Figure 3 will be further discussed later in this section. Following, more details on the results from the workshop is presented.

5.1 Findings from the Artifacts

The SC was originally designed to be used by the developers of informational systems themselves [13]. In our approach, the artifacts were used collaboratively in a participatory practice during the SpW, where researchers and the school's community interacted together. All participants recognized themselves as protagonists in the action of deploying low cost educational laptops at the school.

The Stakeholders Chart. The different groups presented similar results and they are summarized in Fig. 4. On the background, the figure shows part of the model of Stakeholders Analysis [16]; and the table on the foreground shows the main stakeholders elicited.

The *contribution layer* lists those directly involved in the course of action. The core of the analysis was the Project as a whole. Besides those that are part of the

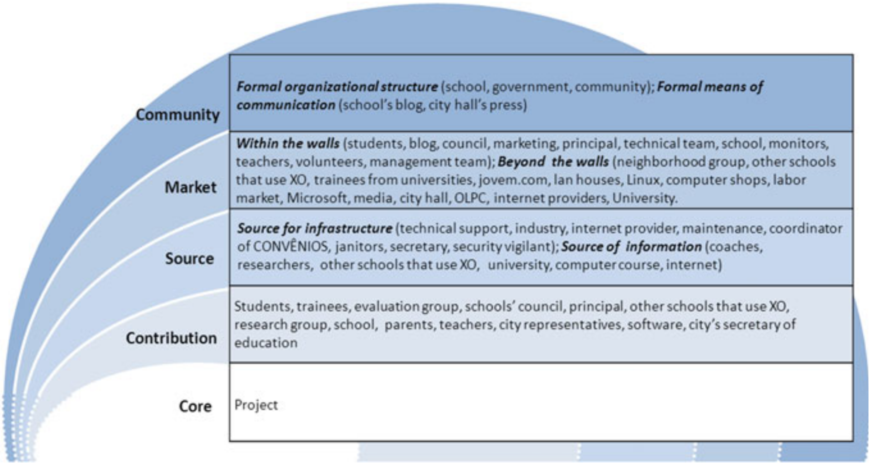


Fig. 4. Summary of the results constructed with the Stakeholders Analysis Chart.

school's community, also other schools were listed on this layer. Those schools that have already been experiencing with the use of educational laptops were seen as possible contributors, as a source for inspiration and example.

For the *source layer*, the list of possible suppliers and clients elicited during the SpW was divided in two categories: those who could be a source of information for the project and those who would provide the necessary material infrastructure.

Also the responses for the *market layer* were grouped in two categories: the collaborators or competitors from within the school's walls, and those from beyond the school's walls. This illustrates how participants are able to see beyond their own and near environment, understanding that the impact of the Project might extend beyond the school itself.

Due to time limits, and also due to the rich discussions raised on these first layers, some of the groups were not able to discuss the outer layer related to *bystanders and community*. Nonetheless, the few responses concerned basically the possible formal means of communication from the community outside the school and other formal organizational structures.

Evaluation Framing. After identifying who were the agents that either would be involved in or affected by the project, the same groups discussed the issues and possible ideas related to each layer of stakeholders. The complete transcription of the issues and ideas raised summed more than five pages of texts. Table 1 lists one example of issue and possible solution related to each layer of stakeholders.

The concern with the distribution of the laptops among students and teachers prevailed in all groups. The school has more than 500 students, plus more than 30 teachers; and there were around 500 laptops available to be shared among them. Different ideas were discussed, including: sharing laptops among siblings or among children from different shifts (at this school, grades 1-5 attends the morning shift, while grades 6-9, the afternoon's).

Table 1. Examples of the ideas elicited by the use of the Evaluation Framing.

Stakeholders	Issues	Ideas and Solutions
Community	What is the government's role in the project?	Technical support from the Municipal Education Secretary
Market	How to provide internet access to the laptops?	Extend wireless network
	Publicity: how to spread the word about the project?	Write reports about the experience; Use the school's official blog.
Source	How to provide training to staff for the usage of the laptops in the school?	Student monitors as assistants
	Maintenance: how to fix software and/or hardware problems?	Provide specific courses for teachers, students and other interested in helping with the maintenance of the laptops; Search for agreements with partners from outside the school
Contribution	Distribution of the laptops: since the amount of teachers/students is greater than the amount of available laptops, who receives the laptops?	Share laptops among siblings; share laptops among children from different shifts
	Children's safety: is it safe for the children to go home carrying a laptop? Would the parents allow it?	Parents signing a Terms of Conditions of Use; Meeting with parents; Parents accompanying children to and from school

Regarding the stakeholders from the source layer – in this case, more specifically, source of information – some participants discussed about the learning curve and who would provide teachers with the necessary training.

5.2 Expectations and Perceptions

The adjectives and sentences written at the end of the workshop portray the beliefs, expectation and perceptions that participants had about the project and the XO laptop.

Fig. 5. Tag cloud formed with participants' adjectives and sentences about the project.

The full transcript of participants' expectations, as well as the transcription of all issues and ideas elicited, shall be available on a technical report.

From the transcription of participants' expectation and perceptions, a tag cloud was created. Figure 5 illustrates the words and their occurrence (how frequently the words appear in participants reports).

As the project was at the core of the discussion, the word "project" was the most mentioned one, and it was taken away from the list so that all other less mentioned words would be readable in the resulting tag cloud. Positive adjectives were also frequently mentioned, e.g., "great" and "good". The general opinions expressed revealed participants' interest and excitement for the new arrival, which was denoted by terms like "interesting", "innovative", "challenging", "motivation", "cool", "fantastic" and "happy".

Albeit naïve, one hoped for changes to "better lives" simply by the presence of the laptop itself. Others demonstrated being aware that the proper use of the laptops might promote deep changes in the school, changing paradigms and impacting the entire school's systems; and they were also aware that this does not happen overnight, but instead, as a result of a long process that demands conjoint actions ("Innovative, however, a long path lays ahead before being totally functional"; "(…) interesting, but demands more detailed planning to work (…)"; "(…) if well organized (…) a mechanism to promote the involvement of the community with the school").

From the 60 participants, two questioned the outcome or purpose of the project. One concern was with the improper use of the internet. The other made use of a metaphor from the bible, questioning whether the project was not trying to "throw pearls to pigs".

5.3 Analysis Summary

The insertion of the laptops that was about to take place at the school was an act that would take place at the technical layer of the Semiotic Onion. This occurrence would demand that norms be created at the school to rule the use of the laptops at the six levels of the Semiotic Framework, as summarized earlier in this paper on Fig. 3.

Social Level. From a social level perspective, one of the rules that needed to be decided regarded sharing laptops: who would share the laptops and how was it going to be controlled. One idea that was positively taken by most of the adult participants was sharing the laptop among siblings. It seems feasible and practical that if brothers and sisters attend the same school, in different grades and shifts, they could perfectly share one laptop. That idea, however, would not work in practice. Students revealed that the relationship they actually have with their brothers and sisters is not always friendly enough to keep that norm working well.

If the norm (formal layer) about sharing the laptops among siblings were decided and implemented without accounting children's opinion, conflicts might had led to a disruption at the social level (the social level of the Semiotic Framework, in this context, can also be seen as the informal layer of the Semiotic Onion).

The involvement of the children in the decision processes are important not only to guarantee that aspects from real live practices are accounted, but also because they – the children – are central to the entire project. The meaning the children, together with other members of the school, will make of the laptops will have social consequences and might determine the success or failure of the project.

As Liu ([16], p. 111) argues: “Before introducing an IT system, there should be clear specifications of rules for business operations”. These rules must make sense to those involved: “an IT system presupposes a formal system, just as a formal system relies on an informal system”.

Pragmatic Level. At a pragmatic level, the use of the laptops should provide richer learning environment, promoting improved education. Another issue raised during the SpW regarded the lack of knowledge teachers had on the laptop. One of the ideas suggested that students who were more familiar with digital technology could provide support to peers and to the teachers. This approach might contribute to an important change: moving away from the instructional paradigm, towards more constructivist/ constructionist [26] ones.

Learning about specific features of the laptop only for the purpose of learning about that technology might make less sense than engaging on the construction of knowledge supported by the use of the technology.

Semantic Level. In this process, the meanings that will be constructed (and reconstructed, in a continuous and iterative process) have an important role. “The meaning of a sign relates to the response the sign elicits in a given social setting”; moreover, it “frequently suggests mental and valuational processes as well” ([16] p. 30). Children might understand the laptops as a source of distraction and recreation. This can be a powerful learning tool if teachers choose to take advantage of what children most like doing at the laptop in order to create learning opportunities. On the other hand, recreational activities might be considered harmful and forbidden

Some of the sentences written at the end of the SpW suggested that it is important to discuss and review pedagogical projects, policies and practices of the school. Indeed the construction of meaning from the uses of the laptop might promote changes which will demand that norms be adjusted to the new reality.

Syntactic, Empiric and Physical Levels. The lower levels of the Semiotic Ladder provide support to the higher levels. The syntactic level houses concerns on properly understanding the rules (of interaction) that allows the use of the laptops. The operational system and the activities (applications) from the XO laptop have interaction metaphors that differ from the ones we usually see in personal computers or regular laptops. These differences might confuse those who are familiar to other computers’ interaction language, but will be overcome as long as the activities are intended to a practical and meaningful use.

The last two levels, physical and empirical, are usually not the concern of users. However, they are important concerns and they were discussed by all groups. Although the XO laptop has the Mesh network that supports communication among the laptops even when no Internet connection is available, participants were concerned with wireless Internet access. Another concern was with the availability of enough electrical sockets to recharge all laptops in the classrooms. Such concerns were easily addressed and solved.

Table 2. Workshops (WS) and objectives.

WS (date)	Main objective of the WS
1st WS (June 2010)	Make it clear to all participants: who are the stakeholders, what are the related problems and possible solutions
2nd WS (July 2010)	Preliminary construction of educational Use Sceneries for the XO laptop [1]
3rd WS (September 2010)	Discussion on strategies for the distribution, management and maintenance of the more than 500 laptops at the school
4th WS (November 2010)	Teachers' development workshop based on the Use Sceneries proposed in the 2nd WS
5th WS (December 2010)	School members' presentations on their experience in the initial use of the laptops; discussion [21]
6th WS (April 2011)	Researchers' presentations on their projects and relation to the school [18], [12]
7th WS (June 2011)	Discussion about metrics, instruments and procedures for the evaluation of the project by the school
8th WS (September 2011)	Collective construction of instruments for the evaluation of the project
9th WS (December 2011)	School members' presentations on their experience during 1 year of use of the laptops; discussion
10th WS (April 2012)	Presentation on use possibilities for the XO laptop; Collective construction of the structure for the book to be written about the XO Project

6 Overview of 'XO Project' Practices

The Workshop (WS) we reported in this paper was the first Semio-Participatory Workshop from the XO Project. After that, other nine Workshops were conducted, as well as other activities within the school community. The purposes of the workshops are summarized in Table 2.

Figure 6 illustrates some of the main activities that took place during the last two years of Project. As indicated in the timeline from Fig. 6, from the donation of the laptops in August 2009 until the beginning of its use by the school in December 2010, many activities took place in order to make sure the use would be meaningful for the community. As in any other organization, the introduction of a new resource in the environment should be planned and organized. As mentioned before, the Workshop reported in this paper happened before the delivery of the laptops at the school.

Until December 2010 the school's building was being remodeled and the school was being "housed" by a nearby university. Even in a strange environment, the laptops were deployed and their use helped students in the process of getting familiar with the new surroundings (e.g., one of the sceneries proposed during the 2nd WS involved taking pictures and writing about the fauna and flora of the school). Even so, relocating to their own "home" made the community feel more comfortable.

As mentioned in Sect. 5.1, the group came up with a solution for the problem of providing teachers with training course for the use of the laptops. The solution was the

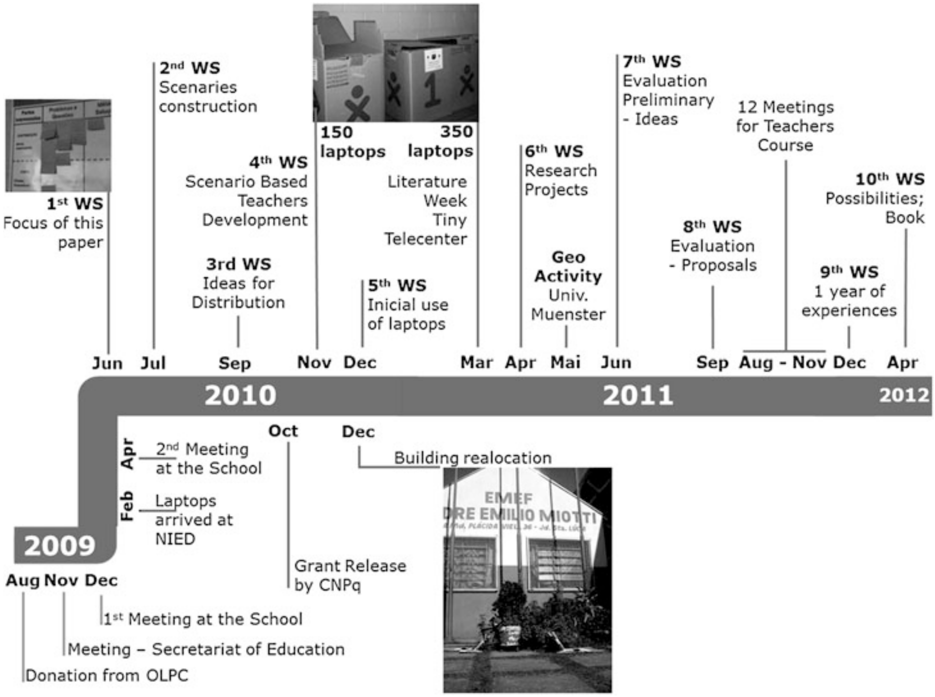


Fig. 6. Time line from the XO Project.

creation of the Students Monitor program, where students would volunteer to support their peers in the use of the laptop.

Even with the program, some teachers asked researchers for specific training. From August until November 2011 we promoted 12 encounters with teachers and some student monitors with the objective of presenting them the main resources of the laptop. The result of this endeavor was very low when compared to the learning that happened "on the fly". When teachers and student monitors used the laptop during regular school classes they had to learn from their own practices mediated by the XO. This seemed to have had more impact than their specific training. The practical use of the laptops allowed them to situate their learning about the features of the laptop in their classes. This result was noticed due to the qualitative research method of participant observation. This qualitative approach, as well as other initiatives and results related to the XO Project, will appear in the book that is being written by the school community and researchers.

7 Conclusions

Schools are complex social organizations that shape the future of generations. The insertion of a technological innovation within the school should not happen with the deterministic belief that technology develops autonomously and by its own logic. The belief should be towards an environment that is continually reconstructed in communicative practices among participants (mediated by technology). The Semio-Participatory approach, grounded in the Organizational Semiotics concepts, helped the group to face the challenge of changing old concepts. One of them is the concept that the school has to adapt itself to technological progress. Instead, the school is an organization that is capable of influencing the technological innovation, inside and outside its walls.

In the initial phase of the XO Project, the first Semio-participatory Workshop sought to clarify the problem and handle with general expectation about the project. The activities, guided by artifacts inspired from OS and PD, allowed the group to line up the different views (viewpoints of researchers, students, teachers and school staff) on the deployment of a new digital technology at an elementary public school in Brazil.

The SpW helped participants to have a broader view of the Project, and to articulate issues, ideas and solutions. Taking a participatory approach to this analysis was essential. This paper described the process of conducting the SpW, illustrating its planning, implementation, results and analysis.

The results indicated that the referential basis and artifacts from OS and PD are appropriate tools for guiding a collective construction of meanings and norms regarding the introduction of a new information technology at an educational organization.

Acknowledgements. We thank CNpQ (processes# 143487/2011, 475105/2010-9 and 560044/2010-0) for financial support; and our colleagues from UNICAMP and EMEF Padre Emílio Miotti for inspiring discussions and support.

References

1. Amiel, T., Arantes, F., Miranda, L., Martins, M.C., Baranauskas, M.C.: A Participatory Approach to Scenario Development for XO Laptops in Brazil. Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications, Lisbon, Portugal (2011)
2. Baranauskas, M.C.C., Schimiguel, J., Simoni, C.A.C., Medeiros, C.M.B.: Guiding the process of requirements elicitation with a semiotic approach In: Proceedings of HCI International, vol. 3, pp.100–111 (2005)
3. Baranauskas, M.C.C.: Socially aware computing. In: Proceedings of ICECE 2009, pp. 1–5 (2009)
4. Bonacin, R., Baranauskas, M.C.C.: Semiotic conference: work signs and participatory design. In: 10th HCI International, pp. 38–43 (2003)
5. Cervantes, R., Warschauer, M., Nardi, B., Sambasivan, N.: Infrastructures for low-cost laptop use in Mexican schools. In: Proceedings of CHI'11, pp. 945–954 (2011)
6. CGI ICT Education 2010: Survey on the use of information and communication technologies in Brazilian Schools. http://www.cetic.br/tic/provedores/index.htm Accessed Feb 2012
7. Corrêa, A.G.D., Assis, G.A., Ficheman, I.K., Venâncio, V., Lopes, R.D.: Avaliação de Aceitabilidade de um Computador Portátil de Baixo Custo por Criança. In: Proceedings of SBIE (2006)

8. Escalona, M.J., Torres-Zenteno, A., Gutierrez, J., Martins, E., Torres, R.S., Baranauskas, M.C.C.: A Development Process for Web Geographic Information System. In: ICEIS 2008, pp. 112–117 (2008)
9. Flores, P., Hourcade, J.P.: One year of experiences with XO laptops in Uruguay. Interactions **16**(4), 52–55 (2009)
10. Furtado, M.E.S., Kampf, T., Piccolo, L.S.G., Baranauskas, M.C.C.: Prospecting the appropriation of the digital TV in a Brazilian Project. Comput. Entertain. **7**, 10–32 (2009)
11. Guimarães, M.S., Baranauskas, M.C.C., Martins, E.: Communication-based modeling and inspection in critical systems. In: ICEIS 2008, v. HCI, pp. 215–220 (2008)
12. Hayashi, E.C.S., Baranauskas, M.C.C.: The affectibility concept in systems for learning contexts. Int. J. e-Learning Secur. (IJeLS) (IJeLS), **1**(1/2), 10–18 (2011)
13. Kolkman, M.: Problem articulation methodology. Doctoral dissertation, University of Twente, Enschede, The Netherlands (1993)
14. Lim, C.P.: A theoretical framework for the study of ICT in schools: a proposal. Br. J. Educ. Technol. **33**(4), 411–421 (2002)
15. Liu, K.: Semiotics in Information Systems and Engineering. Cambridge University Press, Cambridge (2000)
16. Liu, X.: Employing MEASUR methods for business process reengineering in China. Ph.D. thesis, University of Twente, The Netherlands (2001)
17. Liu, K., Sun, L., Tan, S.: Using problem articulation method to assist planning and management of complex projects. In: Project Management and Risk Management in Complex Projects , Part 1, pp. 3–13 (2007)
18. Maike, V.R.L.M., Miranda, L.C., Baranauskas, M.C.C.: Investigando sobre Requisitos para um Jogo de RPG com Professores de uma Escola Pública de Ensino Fundamental. In: SBIE 2011 XXII Simpósio Brasileiro de Informática na Educação, 2011, Aracaju. Anais do XXII SBIE - XVII WIE. Aracaju (2011)
19. Melo, A.M., Baranauskas, M.C.C., Soares, S.C.de M.: Design com Crianças: da Prática a um Modelo de Processo. In: Revista Brasileira de Informática na Educação, vol. 16(1), pp. 43–55 (2008)
20. MEC/INEP. Resumo técnico - censo escolar 2010. http://portal.mec.gov.br/index.php?option=com_docman&task=doc_download&gid=7277&Itemid. Accessed Feb 2012
21. Miranda, L.C., Hayashi, E.C.S., Assis, R., Martins, M.C., Baranauskas, M.C.C.: Laptop XO Dentro e Fora da Sala de Aula: Expressão, Diversão e Alfabetização. In: XXII Simpósio Brasileiro de Informática na Educação (SBIE) & XVII Workshop de Informática na Escola (WIE), pp. 1352–1361 (2011)
22. Moody, L., Schimidt, G.: Going wireless: the emergence of wireless networks in education. J. Comput. Sci. Coll. **19**(4), pp.151–158 (2004)
23. Muller, J.M.: Participatory design. Commun. ACM **36**(6), 26–28 (1993)
24. Neris, V.P.A., Almeida, L.D., Miranda, L.C., Hayashi, E.C.S., Baranauskas, M.C.C.: Collective construction of meaning and system for an inclusive social network. In: IJISSC, vol. 2(2), p.20 (2011)
25. OLPC: http://one.laptop.org/ Accessed in January 2012
26. Papert, S.: The Children's Machine: Rethinking School. In: The Age Of The Computer. HarperCollins (1993)
27. Schimiguel, J., Melo, A.M., Baranauskas, M.C.C., Medeiros, C.M.B.: Accessibility as a quality requirement: Geographic information systems on the web. In: Proceedings of CLIHC'05, pp. 8–19 (2005)
28. Schuler, D., Namioka, A.: Participatory design: Principles and practices. Erlbaum, Hillsdale, NJ (1993)

29. Silva, B.F., Romani, R., Baranauskas, M.C.C.: SOO Brasileiro: Aprendizagem e Diversão no XO. In: RBIE, vol. 16(3), pp. 29–41 (2008)
30. Stamper, R.K.: In Requirements Engineering, Technical and Social Aspects. Academic Press, New York (1993)
31. Stamper, R.K.: A semiotic theory of information and information systems/applied semiotics. In: Invited Papers for the ICL/University of Newcastle Seminar on "Information", pp. 6–10 (1993b)
32. Stamper, R., Liu, K., Hafkamp, M., Ades, Y.: Understanding the roles of signs and norms in organisations: a semiotic approach to information systems design. J. Behav. Inf. Technol. **19**(1), 15–27 (2000)
33. XO Project (2010) XO na Escola e Fora Dela: Uma Proposta Semio-Participativa para Tecnologia, Educação e Sociedade. CNPq, 475105/2010-9 (http://www.nied.unicamp.br/xounicamp/) Accessed 20 Sept 2012

Values and Cultural Aspects in Design: Artifacts for Making Them Explicit in Design Activities

Roberto Pereira(✉), Samuel Bastos Buchdid, and M. Cecília C. Baranauskas

Institute of Computing, University of Campinas (UNICAMP),
Av. Albert Einstein N1251, Campinas–SP CEP 13083-852, Brazil
{rpereira, cecilia}@ic.unicamp.br, sbuchdid@yahoo.com.br

Abstract. Values and cultural aspects must be considered for providing design solutions that make sense to people. Although there is influential literature work devoted to the subject of values and culture in design, there is still a lack of principled and light-weighted artifacts and methods to support designers in this task. In this paper we propose two artifacts for supporting designers in making explicit both stakeholders' values and system's requirements in design. We situate our discussion in a case study where the artifacts were used to support the design of seven prototypes of applications for Interactive Digital Television in the Brazilian context. The results were promising and indicated the artifacts' usefulness to support designers in dealing with values and cultural aspects in design activities.

Keywords: VIF · CARF · Organizational semiotics · VCIA

1 Introduction

Interactive systems are a growing reality worldwide. People use them for different purposes, in quite different and complex contexts, and with unforeseen and far-reaching consequences. They are a clear example of how technology has left the boundaries of offices and workplaces to pervade every aspect of people's personal and social life. As Sellen et al. [1] highlight, as far as people are not just using technology but living with it, values become a critical issue and must be explicitly involved in the design of interactive systems.

As design is an activity no longer confined to specific contexts, several authors, such as Bannon [2] and Cockton [3], have claimed a rethinking of the way interactive systems are designed. For them, it is necessary to focus on the intention of design as a means to improve the world by reimagining, acting, and delivering new sources of value. Winograd [4] had already asserted that the design role "goes beyond the construction of an interface to encompass all the interspace in which people live", requiring a shift from seeing the machinery to seeing the lives of the people using it. According to the author, there is a complex interplay among technology, individual psychology and social communication, in a way it demands attention to relevant factors that become hard to quantify and even identify.

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 358–375, 2013.
DOI: 10.1007/978-3-642-40654-6_22,

Knobel and Bowker [5] point out that conversations and analysis of values in technology usually occur after design and launch. Consequently, most users are faced with design decisions that are undecipherable to them, that do not reflect a respect and understanding to their way of life, their behavioral patterns and values. For the authors, the issue of values often arises in information technologies as disaster needing management.

Designers necessarily communicate values through the technology they produce [6]. In the context of interactive systems, depending on the way the system is designed it will afford behaviors that are intrinsically related to individuals and the complex context in which they are using it [7]. Individuals will interpret and behave over/through the system influenced by their cultural systems (e.g., values, beliefs, behavior patterns). In this sense, as Friedman [6] highlights, although the negligence to values in any organization is disturbing, it is particularly damaging in the design of computer technology, because, unlike the situation where people can disagree and negotiate with each other about values and their meanings, they can hardly do the same with technology. Therefore, understanding the role of human values in technology design is a key factor to the development of technologies that make sense to people and do not produce side effects that harm them.

Miller et al. [8] and Sellen et al. [1] point out values as the critical issue when designing technologies for the digital age. Some authors have explicitly addressed issues on values in technology design. Cockton [3] proposes a framework to support a Value-Centred Design, suggesting activities and artifacts to support designers in an understanding of technology design as a process of delivering value. Adopting a different perspective, Friedman [6] has been working on an approach she named Value-Sensitive Design, to support concerns regarding values, especially the ethical ones, in the design of software systems.

Other authors have investigated the influences and impacts of cultural factors in technology design [9, 10] and other have argued for studies, methods, artifacts and examples for supporting designers to deal with the complexity and different requirements that current technologies demand [8, 11]. Although the previously cited works have shed light on this subject, there is a gap between discussions about values in technology design and practical solutions for supporting designers in this task. Additionally, despite the acceptance of the cultural nature of values, values and culture are frequently approached as independent issues in technology design. To our knowledge, no informed approach or method is explicitly concerned with supporting the understanding and involvement of both values and their cultural nature in the design of interactive systems.

In this work, we draw on Organizational Semiotics (SO) theory [12] and the Building Blocks of Culture [13] to create two artifacts for supporting designers in a value-oriented and culturally aware design of interactive systems. The first artifact, named Value Identification Frame (VIF), supports designers to reason about and list the values related to the different stakeholders that may be direct or indirectly interested and/or affected by the system being designed. The second one, named Culturally Aware Requirements Framework (CARF), organizes the identification of requirements related to cultural aspects that may impact on stakeholders' values. The artifacts were conceived to facilitate their use by professionals that are not familiar with social sciences, and were experienced by 34 prospective designers in the context of seven different projects of social applications for the Brazilian Interactive Digital Television (iDTV). In this paper we present the artifacts, the theories underlying them, and discuss the results obtained from their usage in the practical context.

2 Values, Culture and Interactive Systems Design

Friedman et al. [14] understand values as something that is important to a person or group of people, and Schwartz [15] as desirable, trans-situational goals that vary in importance and that serve as principles that guide people's lives. For Schwartz, values are motivational constructs that transcend specific situations and actions, serving as standards or criteria to guide the selection of actions, policies, people and events.

Values are bound to culture [12, 15] in so subtle ways that people realize they exist usually when rules that impact on them are broken or violated. In many different ways, culture influences on what people pay attention to and what they ignore, what they value and what they do not, the way they behave and the way they interpret other's behavior. The natural act of thinking is strongly modified by culture [16]. In this sense, if we are to approach values in interactive systems design, we must pay attention to their cultural nature and complexity.

When talking about culture, Hall [16] believes it is more important to look at the way things are put together than at theories. Hall [12] introduces the notions of informal, formal and technical levels in which humans operate and understand the world, and approaches culture as a form of communication giving emphasis to the nonverbal. In the OS theory [13], the informal, formal and technical levels are structured in a scheme named "Semiotic Onion" that represents the idea that any technical artifact is embedded in a formal system, which in turn, exists in the context of an informal one. The OS considers an organization and its information system as a social system in which human behaviors are organized by a system of norms. For Stamper et al. [17], these norms govern how members, think, behave, make judgments and perceive the world, being directly influenced by culture and values.

Aiming to formalize and structure the characterization, analysis and comparison between different cultures, Hall [12] proposes 10 Primary Messages Systems (PMS), or areas, named the basic building blocks of culture — see Table 1. According to the author, all cultures develop values with regard to the 10 areas. For instance, values in "Defense" are related to the rules, strategies and mechanisms developed in order to protect the space (physical, personal), the objects used to guarantee protection, the kind of medical therapy adopted/preferred, etc. Values in "Association" are related to the way society structure itself, its groups, public and private organizations/entities, the role and importance of family and other social relationships (e.g., friendship, partnership, marriage), and so on. Values in "Exploitation" are related to the preferred tools, objects, instruments, and procedures for working, playing, learning, protecting, eating, etc.

Values may also be developed in the intersection of different areas and one may approach them in terms of the informal, formal and technical levels. For instance, "Privacy" may be understood as a value developed in the intersection of "Protection"

Table 1. Hall's [12] building blocks of culture.

PMS	Description
Interaction	Everything people do involves interaction with something/someone else: people, systems, objects, animals, etc. The interaction is at the centre of the universe of culture and everything grows from it
Association	All living things organize their life in some pattern of association. This area refers to the different ways that society and its components are organized and structured. Governmental and social structures may vary strongly according to the culture
Learning	Learning is one of the basic activities present since the beginning of life. Education and educational systems are strongly tied to emotion and as characteristic of a culture as its language
Play	Funny and pleasure are terms related to this area. Although its role in the evolution of species is not well understood yet, "Play" is clearly linked to the other areas: in learning it is considered a catalyst; in relationships a desirable characteristic, etc
Defense	Defense is a specialized activity of vital importance. People must defend themselves not only against hostile forces in nature, but also against those within human society and internal forces. Cultures have different mechanisms and strategies of protection
Exploitation	It is related to the use of materials in order to explore the world. Materials in an environment are strongly related to the other aspects of a culture. It is impossible to think about a culture with no language and no materials
Temporality	Time is related to life in several ways: from cycles, periods and rhythms (e.g., breath rate, heartbeat) to measures (e.g., hours, days) and other aspects in society (e.g., division according to age groups, mealtime). The way people deal with time and the role of time in society varies across cultures
Territoriality	It refers to the possession, use and defense of space. Having a territory is essential to life; the lack of a territory is one of the most precarious conditions of life. There are physical (e.g., country, house) as well as social (e.g., social position, hierarchy) and personal spaces (e.g., personal data, office desk)
Bisexuality	It is related to the differences in terms of form and function related to gender. Cultures have different forms of distinction and classification and give different importance to each one
Subsistence	This area includes from people's food habits to the economy of a country. Professions, supply chains, deals, natural resources, are all aspects developed in this area and that vary strongly according to the culture, being influenced not only by the other areas but also by geographical and climatic conditions

and "Territoriality" areas. People from different cultures tend to have their own informal understanding of what privacy is and what it means. There are social protocols, conventions, rules and laws that are formally established to define the meaning, limits and guarantees of an individual's privacy and that varies according to the culture being analyzed. There are also some facets of privacy that are so formally accepted that can be technically supported, such as a curtain to cover a window, the wall for restricting the visibility of a house, and the privacy of medical examinations. According to Hall [12], although humans operate at only one level a time, all the three are always present in any situation.

In the context of interactive systems, the way the value of "Privacy" (or the lack of it) has being handled and supported by applications, mainly the so-called Social Software, has been the cause of several problems widely reported in the Web. Winter [18] draws attention to how *Facebook*® has become a worldwide photo identification database and highlights that privacy issues go from what the application does with users' data to what it allows other applications to do. In the complex scenario of designing interactive systems for wide audiences, designers have to show an understanding of the different ways people value and manage their privacy, and also to comply with the laws established in the social environment these people live. Otherwise, the produced system may trigger undesired side-effects both in the environment it is introduced and on the people living in it.

The OS theory [13] provides methods (e.g., Problem Articulation Method, Norm Analysis Method) and artifacts (e.g., Semiotic Ladder, Ontology Charts) that support designers in considering the social world and its complexity from the articulation of problems stage to the modeling of computer systems. The Stakeholder Identification Diagram (SID) is an artifact from OS — see Fig. 1, that supports the identification of all the stakeholders direct or indirectly affected by the system being designed. The artifact distributes stakeholders into different categories: from the actors directly involved in the project to the people who may not use the system but may be affected by it. The SID considers that each group of stakeholder brings different perspectives to the innovation being designed, having its own cultural system that governs the way it will see, understand, value and react to the proposed innovation [19].

Understanding the way different stakeholders would value and react to an innovation requires designers to see the world through the lenses of these different stakeholders. The Valuation Framing (VF) is another artifact from OS [13] that helps in carrying out this kind of reasoning by favoring the analysis of the cultural dimensions of a product. The VF is built on Hall's [12] areas of culture with a few adjustments. For instance, "Defense" was renamed to "Protection" and "Bisexuality" to "Classification" [19] in order to encompass, beyond the notion of gender, issues of age, instructional, social and economical levels. In the artifact, the analyst's work

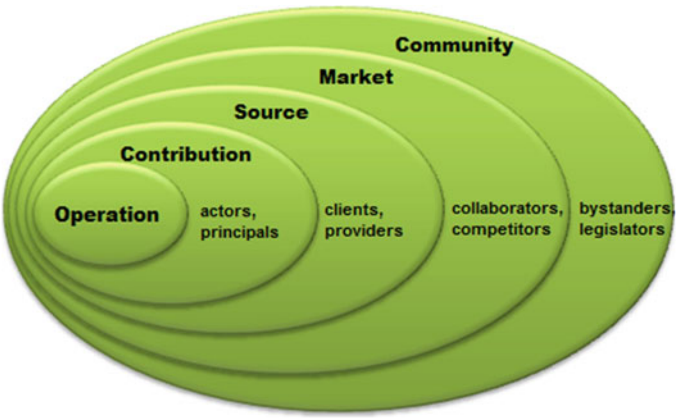


Fig. 1. Stakeholder Identification Diagram (SID). Adapted from Kolkman [19].

consists of questioning, predicting and hypothesizing how the innovation may affect/is affecting the different groups of stakeholders regarding the 10 areas.

3 Two New Artifacts for Supporting Designers

As Sellen et al. [1] suggest, the curricula in Computer Science do not traditionally direct much effort in enabling its students to cope with social issues. It stresses as important the work with multidisciplinary teams that can contribute with different visions to a project. Multidisciplinary teams, however, are not always possible or viable due to project's scope, restrictions and limitations. Consequently, as Miller et al. [8] highlight, if designers working in industrial settings are to account for values, we have to provide them light-weight and principled methods to do so.

We have used artifacts from OS [13] and techniques inspired on Participatory Design [20] to support design activities in different contexts [21]. However, dealing with values is not a trivial activity, and designers need practical artifacts to help them to think of values in an explicit way and to identify the project's requirements related to these values. Following, we present the VIF and CARF artifacts, both created on the grounds of OS theory [13] and Hall's [12] building blocks of culture. The artifacts' templates are available at: www.nied.unicamp.br/ecoweb/products/artifacts.

The VIF artifact was created to support the identification of the values related to the different stakeholders that may be direct or indirectly interested and/or affected by the system being designed — see Fig. 2. Its **input** is the list of stakeholders identified

Project	<Project's name>
Values	<List of suggested values>
CONTRIBUTION	
Stakeholder	**Values**
Stakeholder A	Values related to the Stakeholder A
Stakeholder B	Values related to the Stakeholder B
SOURCE	
Stakeholder	**Values**
Stakeholder C	Values related to the Stakeholder C
(...)	(...)
MARKET	
Stakeholder	**Values**
(...)	(...)
(...)	(...)
COMMUNITY	
Stakeholder	**Values**
(...)	(...)
(...)	(...)

Fig. 2. Value Identification Frame (VIF).

through the SID artifact; and its **output** is a list of the values each different stakeholder brings to the project.

The basic principles of the VIF artifact are: each stakeholder has a set of values that may cause/suffer impact with the introduction of the innovation being designed. The analyst's work is to map what values each stakeholder brings to the project and have to be considered in the design.

The artifact is inspired on the SID — illustrated by Fig. 1. Its header has a space in which designers can put the name of the project — corresponding to the SID's core layer, and a list of values to serve as a start point for the activity. The VIF has also four blocks related to the other layers of SID: “Contribution”, “Source”, “Market” and “Community”. Each block has two columns: in the first one, designers put the stakeholders identified in the respective layer; in the second one, they indicate what values the stakeholder is bringing to the project and must be taken into account. Because the SID induces designers to think of all the stakeholders direct/indirectly involved in the system being designed, by preserving its structure, the VIF leads designers to think of the values of all the different stakeholders making them explicit.

The CARF artifact was created to support the identification and organization of requirements that are related to cultural aspects of the different stakeholders and their values — see Fig. 3. Its **inputs** are: the 10 areas of culture; the stakeholders identified through the SID; and the values mapped for each stakeholder through the VIF. The **output** is a ranked list of requirements that are related to the stakeholders and their values.

PMS	P	Requirements	Value	Stakeholder			
				A	B	C	D
Interaction	#	<requirement 01>	<value>	X			X
	#	<requirement 02>	<value>		X		X
Association	#	<requirement 03>	<value>	X	X	X	
Learning							
(...)							
Subsistence							

Fig. 3. Culturally Aware Requirements Framework (CARF).

The basic principles of the CARF artifact are: values are culturally developed according to the Hall's 10 areas of culture. Depending on the way the innovation is designed, it will impact on different aspects of these areas, promoting/inhibiting the values of different stakeholders. The analyst's work consists of: (i) identifying requirements for the project according to the 10 areas that are related to the values of the different stakeholders, (ii) defining priorities among these requirements, and (iii) dealing with possible conflicts.

The artifact is inspired on the VF. The column "PMS" presents the Hall's 10 areas of culture [12]; the column "P" indicates the priority for each requirement specified (e.g., "3"–High, "2"–Average; "1"–Low); the column "Requirements" describes the requirements related to each area of culture that may impact on stakeholders' values; and the column "Stakeholder" indicates the stakeholders whose values may be positively/negatively affected by the requirement. The column named "Values" serves to make it explicit the stakeholders' value related to each requirement.

In practical terms, the stakeholders identified through the SID, and their values mapped through the VIF are inserted into the artifact, and designers have to reason, make questions and try to identify, in each area, the requirements that are related to the values of these stakeholders. Finally, they mark an "X" in the column of each stakeholder that may be affected by the requirement and assign a priority to the requirement (from 1 to 3).

Depending on the complexity of problem, the number of stakeholders may be very high and listing all the values of all the stakeholders may be an onerous task. In this case, for practical reasons, we suggest designers to consider at least the most representative stakeholders from each layer of SID. This assures that the different forces of information are being represented, reducing the risk of neglecting important issues related to values and culture in the project.

4 Designing for iDTV: A Case Study

In 2003, the Brazilian government instituted the iDTV intending to promote: i) the formation of a national network for distance learning; ii) the access of people to knowledge by reducing economic, geographical and social barriers; iii) the research and development; and iv) the national industry [22]. In this context, values of different stakeholders may suffer and cause influence on the applications, the way they are used, and the impact they may trigger on the society. The government, private organizations, the media etc., have different interests and perspectives regarding the introduction of iDTV in the country. The contents broadcasted, the interaction possibilities, the applications' interface, and even the devices needed for receiving the digital signal and interacting with the iDTV, communicate some of those interests. Brazil is the fifth largest country in territory and population, having a very heterogeneous population in terms of ethnicity, social and economical conditions, and the analogical television is present in more than 97 % of Brazilian homes [23]. Consequently, it becomes critical to think of values and culture when designing applications for the iDTV in order to not deliver applications that trigger undesired side-effects on the society. In this section, we present a practical activity in which the VIF and CARF were used in the design of applications for the iDTV.

The case study was conducted in a Computer Science undergraduate discipline for "Construction of Human–Computer Interfaces", in which the Problem Articulation Method from OS (Liu 2000) was used as an approach for the design of information systems. A total of 34 participants were divided into 7 groups: G1 (formed by the prospective designers: D1, D2, D3, D4 and D5), G2 (D6..D10), G3 (D11..D14), G4 (D15..D19), G5 (D20..D24), G6 (D25..D29) and G7 (D30..D34). The theme proposed to the participants was "social applications for the iDTV". The course took place from August to December, 2011, and by its end each group had to present a functional prototype of its project and socialize the final results with the other groups.

From the 7 projects: G1 and G5 are applications intended to promote sustainable behavior on their users. G2 is an application to support social interaction on football matches programs. G3 and G4 are related to social networks for the iDTV. G6 is an application to support online chat and G7 is related to interactive online courses through the iDTV — see Fig. 4 for some examples. After the course was finished, the groups were asked to voluntarily answer an online questionnaire in order to evaluate the activity and it was requested their permission for using all the material they produced in the course, including their answers to the questionnaire. Another group of 4 participants (G8) opted for not answering the questionnaire and is not being included in this analysis.

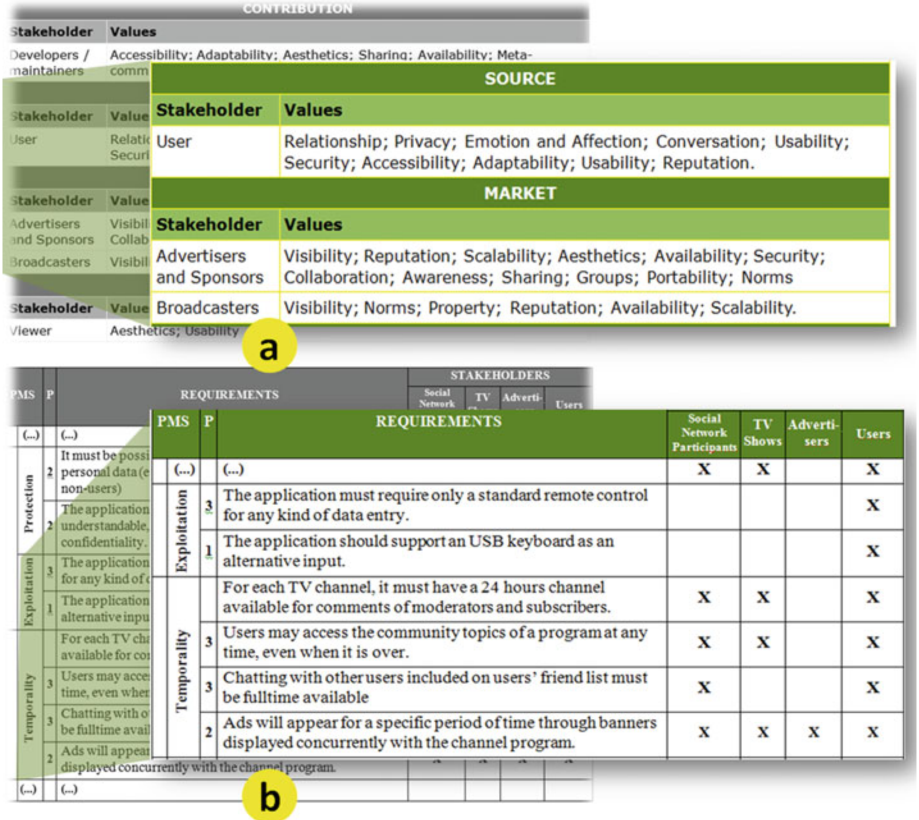
The activity was divided into two parts. In the first part, the groups used the VIF to make it explicit the values each stakeholder was bringing to their projects. In the second part, groups used the CARF to identify what requirements they should pay attention to in order to develop systems that make sense to users and do not cause negative effects on them. When the activity started, each group had defined the focus of its project, had identified the stakeholders using the SID, and discussed the possible problems, solutions and ideas related to each stakeholder using the Evaluation Frame (EF) — another artifact inspired on OS, which organizes the stakeholders according to

Fig. 4. Prototypes from G5, G6 and G1.

the SID's structure and invites designers to reason about the problems and solutions related to each one; — see [24] for more details about this artifact.

The main steps when using the VIF artifact were: 1. Participants selected the most representative stakeholders identified through the SID and inserted them into the VIF's corresponding block. 2. For each stakeholder, participants discussed what values it would bring to the project; what would be important to it and how the system being designed would (should) impact on its values. In order to give participants a starting point, 28 values were suggested in the context of systems for promoting social interaction [25]. As a result, each group had a map showing the different stakeholders and their values — Fig. 5a illustrates part of the VIF filled by G3.

The main steps when using the CARF artifact were: 1. Participants selected at least one stakeholder from each SID' layer, inserting them as a new column into the CARF's "Stakeholder" Section. 2. For each area (PMS), they should identify requirements (resources, norms, quality attributes, functionalities, etc.) that should be considered in the system in order to support the stakeholders' values. 3. Participants should mark an "X" in the column of each stakeholder whose values would be



SOURCE	
Stakeholder	**Values**
User	Relationship; Privacy; Emotion and Affection; Conversation; Usability; Security; Accessibility; Adaptability; Usability; Reputation.
MARKET	
Stakeholder	**Values**
Advertisers and Sponsors	Visibility; Reputation; Scalability; Aesthetics; Availability; Security; Collaboration; Awareness; Sharing; Groups; Portability; Norms
Broadcasters	Visibility; Norms; Property; Reputation; Availability; Scalability.

a

PMS	P	REQUIREMENTS	Social Network Participants	TV Shows	Advertisers	Users
(...)		(...)	X	X		X
Exploitation	3	The application must require only a standard remote control for any kind of data entry.				X
Exploitation	1	The application should support an USB keyboard as an alternative input.				X
Temporality		For each TV channel, it must have a 24 hours channel available for comments of moderators and subscribers.	X	X		X
Temporality	3	Users may access the community topics of a program at any time, even when it is over.	X	X		X
Temporality	3	Chatting with other users included on users' friend list must be fulltime available	X			X
Temporality	2	Ads will appear for a specific period of time through banners displayed concurrently with the channel program.	X	X	X	X

b

Fig. 5. (a) VIF filled by G3; (b) CARF filled by G7. Translations were made by the authors.

promoted/inhibited by the requirement. 4. After filling the artifact, participants should rank the requirements according to their importance to the project.

As a result, each group had a list of requirements related to cultural aspects and values of its stakeholders, a map of the possible impact of these requirements on different stakeholders, and an indication of priority for each requirement — Fig. 5b illustrates part of the CARF filled by G7.

As background material for supporting the activity each group was supplied with: (i) guidelines explaining the activity's steps; (ii) the VIF and CARF artifacts both in press and digital format; (iii) a table containing the list of 28 values in the context of social applications [25]; (iv) a simplified explanation of each area of culture — as in Table 1; and (v) at least 3 questions related to each area the groups should think about — see Table 2. The letters into the brackets in Table 2 indicate the stakeholders

Table 2. Questions in each area for the iDTV context.

PMS	Description
Interaction	What interaction possibilities will the application offer? [D]; What kinds of actions can users perform? With what\who? Why? Through which devices? [U, T]; How do people interact with the analogical TV? What will be changed? [T, U]
Association	Is the application usage individual or collective? [U]; Is there any dependence on other organizations/entities (e.g., data supply)? [S]; May it cause impact on any aspect of collective life? [G, U]; Is it associated with television content? [S]
Learning	Is it required any prior knowledge for learning how to use the application? What is the cognitive effort for learning it? What kind of learning it can provide? [U]; It is required training, new abilities or tools for developing the application? Which ones? [D]
Play	What kind of emotions the application may/should evoke/avoid (e.g., fun, challenge, warning)? Why? [D, G, U]; How the application has to be designed to promote/inhibit these emotions? [D]; What are the possible impacts on users? [U]
Defense	Can the application compromise users' safety? [U]; What are its policy and terms of use? [D, G, S, U]; Is there any rights, patent or property? [G, S, T]
Exploitation	What are the physical devices required to interact with/through the application? [D, T]; Is it required any other material or modification in the environment (e.g., sound, media)? [D, U]; Will the introduction of new devices generate the disposal of old ones? Is there any way to reuse? [D, G, S, U]
Temporality	Is there a formal period for interacting (morning, lunch)? [D, G, T]; What is the expected frequency of use (daily, monthly)? [U]; What about the interaction duration? Is it brief, medium or long? [D]
Territoriality	In which space the application will be used? [U]; Are there specific requirements for the interaction space (size, lighting, sound)? What kind of impact may be generated? [D, U]; Is the usage individual or collaborative? [D, S]
Bisexuality	Are the technologies necessary to develop the application open source? [D]; Is its final cost (including the physical devices) viable/accessible for the different socio-economic conditions of users? [U, G, S]; May it cause negative impact on economic issues? How? [U]
Subsistence	What is the target audience? [U]; Is it required minimum age to participate? [A, G, U, S]; Is it required information redundancy (the same information in different formats)? [D, G, S, U]

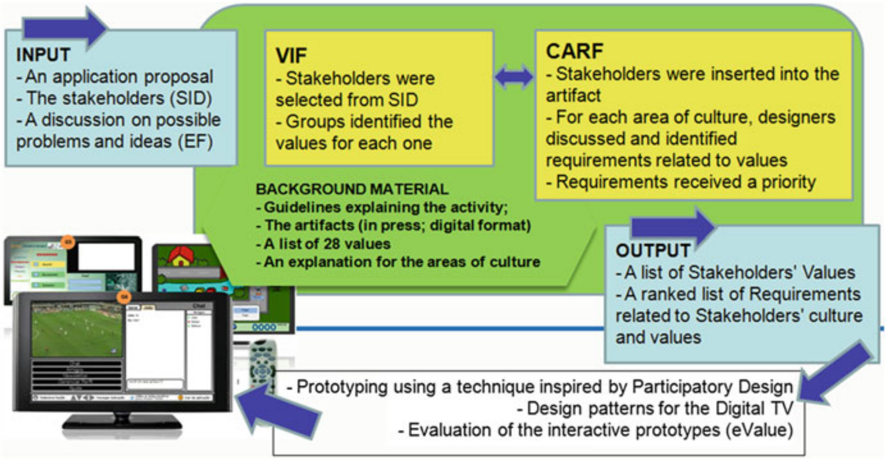


Fig. 6. Methodology of the case study.

directly related to each question: [D] Designer, [G] Government, [S] TV Station, [T] Transmission Industry, [U] User.

The material produced in this activity was used to support groups in the forthcoming steps of their projects. 1. With the list of values and requirements at hands, each group produced the first version of its system's prototype — an adapted version of the Brain Drawing technique [20] was conducted and the iDTV design patterns from [25] were followed. 2. The *Balsamiq*® tool was used to draw the users' interfaces and the *CogTool*® was used to create the interactive prototypes. 3. The eValue artifact [26] was used to evaluate the prototypes according to the way they were supporting the values and cultural aspects critical to each project. Figure 6 illustrates an instantiation of the methodology [28, 29] used in the discipline to conduct the case study.

5 Results

Based on the material produced in the case study, including the final prototypes created by the groups, it was possible to identify the VIF and CARF as promising artifacts for supporting designers in a value-oriented and culturally aware design. Both the artifacts met the needs that led to their conception: (i) thinking of values in an explicit way and (ii) identifying the requirements related to these values.

As an illustration, Fig. 7 shows the prototype produced by G3 regarding a social application for the iDTV. Through the VIF, the group made explicit the values of the stakeholders involved in the project. For instance, the group pointed out "Privacy", "Accessibility", and "Relationship" as values of the stakeholder "users". Through the CARF, the group discussed about the project according to each area of culture, and specified requirements that should be considered in order to account for the values.

Fig. 7. Prototype designed by G3: a social network for the iDTV.

For promoting the value of "Privacy", in CARF's "Protection" area, the group specified that: 1. "Users have to agree explicitly for letting their profile publicly visible". 2. "The application must be included in the 'Parent's Control' functionality, protected by a password". 3. "The application must allow users to turn on/off the 'History recording' feature". The detail (1) in Fig. 7 represents the configuration feature that allows users to choose: (i) whether their activity history will be recorded; (ii) whether other users are allowed to see their updates; and (iii) whether they want to receive recommendations from other users.

For promoting the value of "Accessibility", in CARF's "Exploitation" area, participants specified that the application must have: 1. "The possibility of changing the size of interface elements and the color contrast". 2. "Subtitles for spoken communication". 3. "A help section and additional information about the features". The detail (2) in Fig. 7 indicates the possibility of changing the size of the interface elements and the detail (4) indicates a "Help" feature — it is related to the "Learning" area. Understanding the "Exploitation" and "Learning" areas of culture is key to design an accessible solution in the proposed scenario because, as Neris et al. [30] argue, designers need to know users in their abilities, preferences, and motor and cognitive limitations, formalizing the interaction requirements and investigating solutions of interaction and interface for the diversity. This is very different from developing applications for the "average user" that would not capture the reality of a plural context such as the Brazilian one.

For promoting the value of "Relationship", in CARF's "Association" and "Interaction" areas the participants specified that: 1. "It must be possible for users to interact with each other through chat and messages". 2. "The application should recommend 'friends' to users according to the information of their profile". 3. "It must be possible for users creating their lists of friends, family members, other groups, etc.". The detail (3) in Fig. 7 indicates the feature for managing "friends". Furthermore, we can point out another example: through the VIF, the G3 identified the value of "visibility" for the stakeholder "Sponsorship". In CARF's area of "Subsistence", G3 adopted the strategy of providing ads services for funding the maintenance costs: "The profit will be generated through ads from sponsors and the TV programs". The detail (5) in Fig. 7 indicates a banner where ads are displayed.

Values of other stakeholders and their related requirements were also considered by G3. For instance, "Reputation" is a value of the stakeholder "TV Station" and is related to the area of "Classification". The group specified requirements and designed a feature in which users can rate programs, add them to their favorite list, and share the list with their friends. The same was identified on the projects of other groups. For instance, before using the artifact, G1 (designing a game for sustainable behavior) was not paying attention to the value of "Identity" of its stakeholder "user". When discussing the area of "Classification", participants perceived that their initial ideas would lead to a biased design in which users would have to use the avatar of a little boy — no possible changes were possible. After filling the artifacts, they designed a feature where users could choose between a little boy and a girl avatar, accounting for the differences of gender and preferences when playing.

6 Discussion

According to the answers in the evaluation questionnaire, identifying the values of the stakeholders involved in the application being designed led the groups "*to evaluate the impact of the project on each stakeholder and, then, to adapt the project according to the stakeholders' needs and values*" [G4].[1] Other group mentioned that thinking of values "*contributes to have a wider perception and understanding of the stakeholders involved in the project, their point of view, and the real purpose of the application we should develop to them*" [G5]. And also, that thinking of values "*is of critical importance because it helps us to see who may be affected by the project, and what values we should pay attention to in order not to cause negative side-effects on any stakeholder*" [G6].

Regarding the utility of the VIF and CARF artifacts, groups were asked about their perceived utility and contribution to the project. Two groups answered that both artifacts contribute strongly and were determinant to the identification of the values (VIF) and the requirements related to stakeholders' values and culture (CARF). Four groups answered they contribute to the process, and a group answered they are indifferent (neutral). None answered the artifacts do not contribute or make the activity difficult — see Fig. 8.

For G2, understanding culture and values is mandatory when designing applications for a wide and complex context like iDTV. For G3, this understanding favors "*the identification of important points during the design stage*" preventing re-work,

[1] All translations were made by the authors.

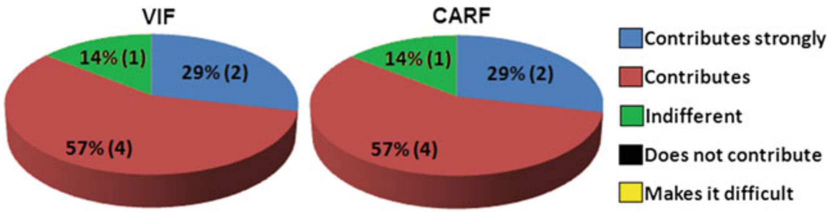


Fig. 8. Contribution of the artifacts to the projects.

additional costs with modifications and even the project's failure. For G4, the artifacts "*contribute to structure and organize ideas*"; they "*support a better view and understanding of the project*", and they "*contribute to the development of the application taking into account the points that are truly important in the users' context*".

When asked about the positive aspects of both artifacts, G1 answered they "*provide a wide perception (what is needed and why), and a basis for reasoning about the project*". G2 cited that the artifacts contribute to "*structure, organize and better understand the ideas for the project*". G4 pointed out that the artifacts are "*simple and easy to understand*" and that they "*direct the project toward the consideration of values*". And G6 answered that the artifacts contribute to "*manage and develop the project, respecting the values of each stakeholder and finding new requirements to the project*". On the other hand, when asked about the negative aspects, G4 asserted that the artifacts "*need additional information for supporting their usage*". G7 cited the high quantity of terms and aspects to be considered. And G2 suggested that the "*areas of culture in CARF could be more explained*" and that the artifacts have "*too many variables, making it difficult to keep the simplicity and to think of only a few stakeholders and their values*".

These aspects suggest that the artifacts must be as simple as possible in order not to overload designers with complex terms and unnecessary steps. However, as the authors we cited previously have argued, dealing with values and culture in technology design is a great challenge we are facing in the present. In part, it is due to the topic's inherent complexity, and that becomes even more difficult due to the lack of training and familiarity with social subjects students in technological areas have. Therefore, some initial difficulty in learning how to use the artifacts is expected.

Indeed, our main concern when creating the artifacts was to find a balance between making them self-explanatory and informative, while keeping them as simple and easy to use as possible. For instance, during the case study we identified that it would be useful to include a column named "Value" in the CARF (see Fig. 3) in order to make explicit the relationship among the requirements, the areas of culture and the stakeholders' values. Additionally, the values from [25] included into the VIF artifact have been used in different contexts [7, 21] and seems to be a good starting point for the discussion on values in applications intended to promote social interaction. In the evaluation questionnaire, groups were asked whether the suggested values contributed to the activity. Two groups (28 %) answered they were indifferent; while three groups (44 %) answered they contributed, and 2 groups (28 %) answered they contributed strongly to the activity — see Fig. 9a.

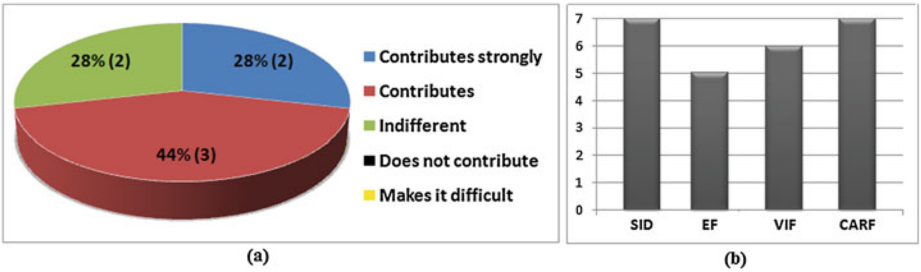


Fig. 9. Contribution of the values (a); Artifacts groups would use in other contexts (b).

For the CARF artifact, groups were asked whether the description of each area of culture, and the questions related to it, contributed to the clarification of requirements related to stakeholders' cultural aspects that could impact on their values. The 7 groups (100 %) answered positively (the artifact contributed), and highlighted that the CARF "*is comprehensive, and the questions make it self-explanatory*" [G1]; "*give a direction in the requirements identification activity*" [G3], and "*it is a well-synthesized structure to support seeing and understanding culture during the development stage; they make you reason on all the aspects that can influence in the project development*" [G4].

Regarding all the artifacts used in the case study, the 7 groups (100 %) answered they would use the artifacts to support their activities in other contexts, mainly when designing a new product to be used by a wide audience. The SID and CARF were cited by the 7 groups (100 %); while 6 groups cited the VIF (86 %) and 5 groups cited the EF (72 %) — see Fig. 9b. It is interesting to note that even the group that was indifferent to the use of the artifacts in the process (see Fig. 8), reported they could use the CARF in another context.

In sum, although further exposition of the artifacts to other students and professional designers in different contexts is still needed, the results obtained from the case study as well as the answers to the evaluation questionnaire indicate both VIF and CARF as promising artifacts for supporting designers in the complex scenario of designing value-oriented and culturally aware solutions.

In another study [21], values were used in the design of applications for supporting cross-cultural collaboration and our findings indicated the values as determinant for guiding designers in the evaluation as well as redesign of their prototypes. The interested reader may consult [27] for detailed information about the evaluation activity supported by the eValue artifact (mentioned in Fig. 6), and for the results that reinforced the contribution of considering values and cultural aspects in all the design stages.

7 Conclusions

Designing technologies that reflect an understanding and respect to human values is an ethical responsibility, a need and a challenge for all those who are direct or indirectly involved with design. However, although clearly recognized as important, there are few initiatives in literature relating culture and values to technology design. There is also a lack of approaches, methods and artifacts for supporting designers in dealing with values and cultural aspects in practical contexts — and even a lack of practical examples to inspire designers in similar situations. In this paper, we shed light on this scenario proposing the VIF and CARF artifacts and suggesting other existing artifacts (e.g., SID, VF, EF) that may support designers in practical settings.

The artifacts were used by 34 prospective designers in a case study related to the design of applications for the Brazilian Interactive Digital Television. The results obtained from this case study indicate the benefits of using the artifacts for supporting designers in keeping values in mind during the design activities, and in identifying requirements related to the cultural aspects of stakeholders that may impact on their values. The case study also suggested some points that could be improved in the artifacts and that may be subject of further studies.

Finally, although the artifacts have shown interesting results, they alone are not enough to guarantee an effective consideration of values and culture in interactive systems design. Indeed, as the case study presented in this paper has shown, other artifacts, methods and tools are needed in order to allow the articulation and involvement of values and other cultural aspects during the different stages of a system design. We are naming value-oriented and culturally informed approach (VCIA) such set of artifacts and methods we are investigating in ongoing and further research.

Acknowledgements. This research is partially funded by CNPq (#560044/2010-0 EcoWeb Project) and FAPESP (#2013/02821-1). This paper is a revised version of a previous paper published at ICEIS 2012: "Keeping Values in Mind: Artifacts for a Value-Oriented and Culturally Informed Design", which received the Best Student Paper Award in the Human–Computer Interaction area. The authors specially thank the editors for the invitation, and the participants of the case study who voluntarily collaborated and authorized the use of the documentation of their projects in this paper.

References

1. Sellen, A., Rogers, Y., Harper, R., Rodden, T.: Reflecting human values in the digital age. Commun. ACM **52**(3), 58–66 (2009)
2. Bannon, L.: Reimagining HCI: toward a more human-centered perspective. Interactions **18**(4), 50–57 (2011)
3. Cockton, G.: A development framework for value-centred design. In: Proceedings of the CHI 2005 Extended Abstracts, pp. 1292–1295. ACM press (2005)
4. Winograd, T.: The Design of Interaction. In Beyond Calculation: The Next Fifty Years of Computing. Copernicus, pp. 149–161. Springer, New York (1997)
5. Knobel, K., Bowker, G.C.: Values in design. Commun. ACM **54**(7), 26–28 (2011)
6. Friedman, B.: Value-sensitive design. Interactions **3**(6), 16–23 (1996)
7. Pereira, R., Baranauskas, M.C.C., Almeida, L.D.: The value of alue identification in web applications. In: IADIS International Conference on WWW/Internet (ICWI), pp. 37–44 (2011)

8. Miller, J., Friedman, B., Jancke, G., Gill, B.: Values tensions in design: The value sensitive design, development, and appropriation of a corporation's groupware system. In: ACM GROUP'07, pp. 281–209 (2007)
9. Del Gado, E., Nielsen, J.: International Users Interface. Wiley, New York (1996)
10. Marcus, A.: International and intercultural user interfaces. In: Users Interfaces for All: Concepts, Methods and Tools, pp. 47–63. Lawrence Erlbaum, Mahwah (2001)
11. Harrison, S., Tatar, D., Sengers, P.: The three paradigms of HCI. In: AltCHI – CHI'07, pp. 1–21 (2007)
12. Hall, E.T.: The Silent Language. Anchoor Books, New York (1959)
13. Liu, K.: Semiotics in Information Systems Engineering. Cambridge University Press, Cambridge (2000)
14. Friedman, B., Kahn, P.H., Borning, A.: Value sensitive design and information systems. In: Human-Computer Interaction and Management Information Systems: Foundations. Armonk, pp. 348–372 (2006)
15. Schwartz, S.H.: Basic human values: their content and structure across countries. In: Values and Behaviors in Organizations. Vozes, pp. 21–55 (2005)
16. Hall, E.T.: Beyond Culture. Anchor Books, New York (1977)
17. Stamper, R., Liu, K., Hafkamp, M., Ades, Y.: Understanding the role of signs and norms in organisations – a semiotic approach to information systems design. J. Behav. Inform. Technol. **19**(1), 15–27 (2000)
18. Winter, J.: Pedophiles find a home on wikipedia. fox news. http://www.foxnews.com/scitech/2010/06/25/exclusive-pedophiles-find-home-on-wikipedia/ Accessed 20 Jan 2012
19. Kolkman, M.: Problem articulation methodology. Ph D thesis. University of Twente, Enschede (1993)
20. Schuler, D., Namioka, A.: Participatory Design: Principles and Practices. Lawrence Erlbaum Associates, Hillsdale (1993)
21. Pereira, R., Baranauskas, M.C.C.: Valuation framing for social software: a culturally aware artifact. In: 13th International Conference on Enterprise Information Systems (ICEIS), pp. 135–144 (2011)
22. Brasil, Presidential Decree No 4.901. http://www.planalto.gov.br/ccivil_03/decreto/2003/d4901.htm (2003) Accessed 20 Jan 2012
23. IBGE. National Survey by Household Sample. http://www.ibge.gov.br/home/download/estatistica.shtm (2010) Accessed 20 Jan 2012
24. Baranauskas, M.C.C., Schimiguel, J., Simoni, C.A.C., Medeiros, C.M.B.: Guiding the process of requirements elicitation with a semiotic approach. In: 11th International Conference on Human-Computer Interaction, pp. 100–111. Lawrence Erlbaum (2005)
25. Pereira, R., Baranauskas, M.C.C. Silva, S.R.P.: Softwares Sociais: Uma Visao Orientada a Valores. In: IX Brazilian Symposium on Human Factors in Computer Systems (IHC'10), pp. 149–158, October 2010
26. Kunert, T.: User-Centered Interaction Design Patterns for Interactive Digital Television Applications. Springer, Heidelberg (2009)
27. Pereira, R., Buchdid S.B., Miranda, L.C., Baranauskas, M.C.C.: Considering values and cultural aspects in the evaluation of interactive systems prototypes. In: International Conference on Information Society (i-Society 2012), pp. 395–400 (2012)
28. Baranauskas, M.C.C., Bonacin, R.: Design – indicating through signs. MIT Press Design Issues **24**, 30–45 (2008)
29. Baranauskas, M.C.C.: Socially aware computing. In: ICECE 2009 VI International Conference on Engineering and Computer Education, pp. 1–5 (2009)
30. Neris, V.P.A., Baranauskas, M.C.C.: End-user tailoring: a semiotic-informed perspective. In: ICOS' 2007 International Conference on Organizational Semiotics, pp. 47–53 (2007)

Composing Interface Demonstrations Automatically from Usage Logs

Tamara Babaian(✉) and Wendy Lucas

Bentley University, Waltham, MA 02452, USA
{tbabaian, wlucas}@bentley.edu
http://cis.bentley.edu/ERP

Abstract. The benefits of Enterprise Resource Planning (ERP) systems for managing enterprise-wide processes and resources may be counterbalanced by the burdens placed on their users in learning to operate the system. In this paper, we present an approach to creating system tutorials in which the system itself produces interface demonstrations as dynamic visualizations of previously occurring system-user interactions. This approach is motivated by our observations of ERP users in the workplace in conjunction with theoretical accounts of system-user collaboration. The playback interface culls information from the usage log records of the data model, which was specifically designed to overcome the challenges of identifying and aggregating relevant process-related data. This approach is a less costly, more flexible alternative to pre-recorded tutorials and has far-reaching implications for assisting users in learning to navigate and use any complex or confusing interface.

Keywords: Human-computer collaboration · Tutorial systems · Enterprise systems · ERP · Human-computer interaction · Usability

1 Introduction and Motivation

Enterprise Resource Planning (ERP) systems integrate internal and external data from across the organization and enforce the best practices of a variety of industries. While this approach can lead to greater efficiencies and lower risks for the company, the enforcement of best practices can be particularly burdensome for the systems' users. Because they are designed to be industry- rather than company-specific, users are confronted with overly complex, opaque interfaces that make it difficult to discern which components and screens are actually relevant for their particular purposes and how they are related to the underlying processes. Those processes are in themselves complex, comprised of multiple tasks involving numerous steps. It is difficult if not impossible for typical users to determine where the tasks they perform fit into the overall process and how those tasks interact with those performed by other users of the system.

Industry reports [1–3] and our own field studies of ERP usage [4,5] have confirmed how difficult it is for users to learn how to perform their tasks using these systems. Even after undergoing months of training, users are often ill-equipped to undertake new tasks or tackle ones they perform on an infrequent

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 376–392, 2013.
DOI: 10.1007/978-3-642-40654-6_23,

basis. The help functionality is typically too generic to be useful and user manuals are similarly unhelpful. As a result, users prefer to turn to their colleagues and in-house experts for assistance.

In this paper, we present an alternative, less resource-intensive approach to providing users with the guidance they are seeking. We have developed a playback interface that creates on-request, dynamic visualizations of system-user interactions for supporting ERP users in learning (or recalling) how to perform a task or a multi-task business process. The system retrieves data from past interactions that have been captured to a usage log and couples it with its own knowledge of its task interfaces in order to create step-by-step playbacks of how users have previously performed a particular task or series of tasks comprising a process. This approach offers greater flexibility and user-control than pre-recorded tutorials and is more likely to be relevant to the context of the user's current interaction with the system.

The motivation for a playback approach to sharing system know-how grew out of the application of the collaborative view of human-computer interaction [6–8] to addressing ERP usability. This perspective holds that the computer must act as a partner by supporting its users in the increasingly complex environments of modern applications. In earlier work, we defined four principles based on this perspective for guiding the design of ERP systems [9]. The playback agent described here is an approach to implementing the second design principle, namely:

> The system should provide navigational and progress guidance to a user performing a transaction, indicating the broader context of each interaction in terms of the related business process components and specifying the completed and remaining parts. A sufficiently competent user should be able to turn off this guidance if it becomes a distraction.

A study of CoScripter [10], a knowledge sharing system, showed that users benefit from using a system that can execute pre-recorded sequences of steps for achieving their daily work goals. The interface presented in this paper implements a similar type of playback behavior within an ERP system.

The major challenge behind composing such playbacks of user-system interactions from vast amounts of usage data comes from the need to identify only those user sessions that are relevant to a process and to aggregate the key-press level data from those sessions [11] in *real-time*. The sessions typically involve multiple users who have worked on parts of the process during non-continuous time periods that may be overlapping. Since the logging facilities of existing ERP systems are not sufficient for this task, we have designed a representation of the tasks, their composition into user interface components, and their relationship to domain objects, which we call the Task-Interface-Log (TIL) model [12]. This model and the algorithms we have developed for enabling playback have been implemented in an ERP prototype that is used here for demonstrating our approach.

Enabling automated demonstrations of system-user interactions supports the common practice of employees learning from others in the organization about how to use the system, which can vary significantly from the prescribed processes [13]. Other benefits include:

- Time and cost savings - users can get the help they need without leaving their desks. The composition of demonstrations is done automatically by streamlining the previously captured record of a user performing a task, without the need for any help desk personnel.
- Flexibility - users can be given choices concerning the interactions they would like to view, including the task or process, the relevant organizational unit, a date range of interest, etc. This flexibility supports organizational memory and maintains the currency of the tutorials, as they are generated in real time from data that includes the most recent instances of business processes performed within the system.
- In-context help - unlike the help functionality currently provided by ERP systems, which is typically too generic to be useful, our demonstrations capture the context of the system-user interaction and are accessible directly from the system interface.

The next section of this paper discusses related work. In Sect. 3, we highlight components of the TIL model that are essential to task and process instance identification. The playback interface is described in Sect. 4. Conclusions and directions for future research are presented in Sect. 5.

2 Related Work

Improving the usability of ERP and other business applications receives scant attention in the human-computer interaction literature, although challenges have been acknowledged. Recently, Quast and Handel [14] have proposed a novel architecture that treats user-tailored interfaces as a dominant design component. It leverages social computing methods for community-driven development and selection of interface instances that best capture the needs of a particular organizational unit. Interestingly, this work implements another one of our design principles [9], which requires enterprise systems to have a mechanism for easy customization of end-user interfaces in order to match the practices and vocabulary of each individual organization.

In acknowledging the need for dramatic improvements in tutorial and help methods, it is noted in [15] that recorded demonstrations of interfaces are a very effective method for helping users learn procedural tasks. The tutorials they describe, however, are prerecorded videos that are accessed via external websites and are not integrated with a user interface. In our approach, automated tutorials are dynamically generated from logs of actual usage, thereby bypassing expenses associated with prerecording videos while providing up-to-date, in-context demonstrations of any process that has been performed with the system.

There are several techniques for providing in-context help with complex interfaces. The focus of many of these approaches is on the interface components themselves. ToolClips [16] helps users understand how to use a tool or function by augmenting traditional textual tooltips with video and documentation content. AIMHelp [17] provides dynamic, context-sensitive help for the AIM (Auckland Interface Manager) GUI-based application, including an automated help index, a browsable event log for listings of prior events, and displays for the causes of event triggers and the effects of a selected widget. The CogentHelp prototype tool views the help system as "one topic per widget" [18], with human-generated snippets attached to widgets in user interfaces built using the Java Abstract Window Toolkit. A creation tool for providing contextual help for GUIs using screenshots is presented in [19].

Various approaches exist for guiding user interactions through a process. The SmartAidè tool [20] provides context-sensitive help to novice users of complex interfaces based on AI planning. It gives step-by-step textual instructions, automatically generates action sequences for execution within the workspace, and changes the state of the interface. The CoScripter tool [10] uses a programming-by-demonstration approach for capturing knowledge on how to perform web-based procedures. Users record their actions as editable, executable scripts that are stored in a wiki for maintaining and sharing among users.

A model-based approach is described in [21], in which a hypermedia interface provides adaptive diagnostics and interactions to maintenance technicians of complex equipment that are tailored to the competency of the user. The OWL (Organization-Wide Learning) recommender system [22] uses data logged from users of instrumented versions of software, with individualized coaching in the form of recommended learning tips that are computed on a monthly basis. In [23], workflow models are applied for assisting users of desktop applications, with an intelligent workflow assistant automatically keeping track of ToDo lists and automating part of the workflow when possible.

In the ERP domain, process mining is typically used passively for workflow analysis. Its active use in an online setting is promoted in [24], which proposes the mining of partial cases that have not yet completed for use in recommending the remaining steps to take for minimizing cost and time. The approach described in [25] provides recommendations on the most suitable next steps from previous processes executed by the current user along with process decisions from all users involved in the same process type. Their self-adjusting user model classifies each user and weighs recommendations based on those classifications. In [26], the focus is on aiding system administrators of large-scale applications by identifying from system logs those choices performed most recently in the past that led to a desired outcome.

While our approach is also model- and process-driven, it is differentiated by our framework, which supports the automatic derivation of processes from usage logs based on records of task inputs and outputs rather than on temporal data. This reduces noise arising from multiple users working on interconnected, highly concurrent processes and avoids many of the shortcomings identified in [27].

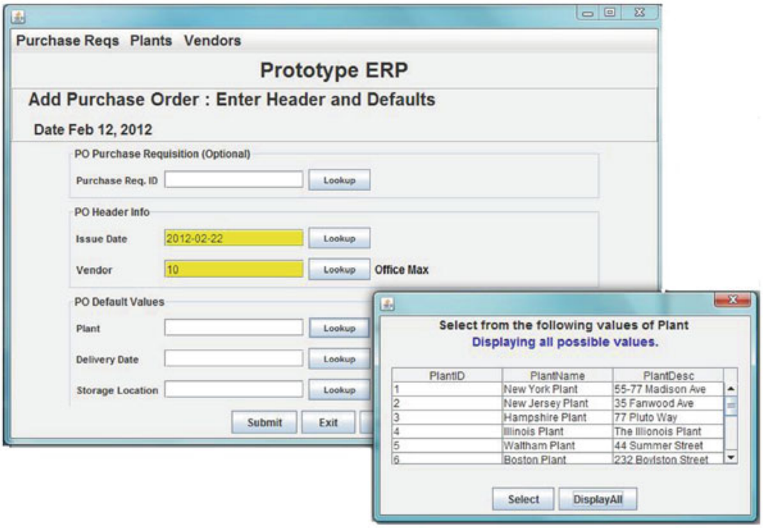


Fig. 1. A screenshot demonstrating typical features of our prototype's interface.

3 Capturing the Usage Data

For the purpose of our investigations, we have developed a prototype that implements a selection of typical ERP tasks. We briefly outline the components of its interface to provide context for the presentation of our model and algorithms that follows.

Each task interface is implemented using one or more interface pages. The graphical user interface of each page uses groups of interactive components that are designed specifically to automatically record all interactions with the user. These interactive components are input fields, standard buttons, and menus.

While the user input components can be laid out in different fashion on different pages, all user-system interactions occur via these input controls. The composition of each page is described within the data model of the system, as discussed next.

3.1 Task-Interface-Log (TIL) Model

We deploy a Task-Interface-Log (TIL) model at the core of our prototype to enable the recording and reconstruction of the details of any interaction between the system and its users. An entity-relation diagram of the essential components of the TIL model presented in Fig. 2 illustrates what follows here.

Our model captures the essential description of the tasks that the system is designed for in the **Task module**. Each task description includes the specification of the type of object produced as the result of executing a task, which is described in an attribute called *DTableOut*. This is an abbreviation for Domain

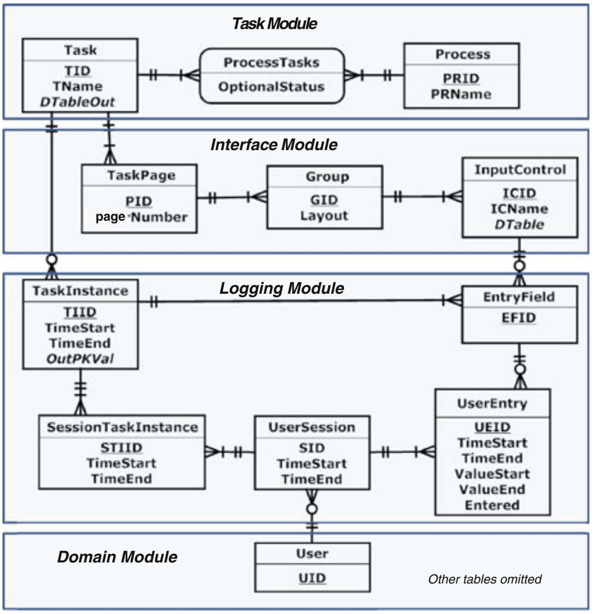


Fig. 2. The Task-Interface-Log (TIL) model.

Table Output, since task output is always stored in a table from the ERP domain database that we call the **Domain module**. For example, the Add Material task results in the creation of a new material record, which is stored in the Material table. Thus, the value of *DTableOut* for Add Material is Material. The Task module also includes user-configurable information regarding the tasks that are included in the business processes.

Each task is implemented as a set of ordered interface pages containing user input components, such as input fields, buttons, and menus. The description of each task page is stored in the **Interface module**. Each input component record in the InputControl table includes a *DTable* attribute, which specifies if the data entered into the component must have a corresponding record in a particular domain table (i.e. a Domain module table). For example, a text field for entering plant values will have the *DTable* attribute set to Plant. Note that a collection of *DTable* attributes of all input components associated with a task specifies the task's *domain inputs*, i.e., the types of domain objects used to produce the output of the task.

The Task and Interface modules are *static*, in that their contents do not change after the system has been configured. They provide the means for creating an interface from a declarative specification as well as for identifying the possible input-output connections between tasks. These connections play an important role in identifying which task instances are potentially related to each other and which are not.

The **Logging module** records user interactions with the system on two interconnected levels: the task level and the interface level. Logging on the task level involves keeping track of *task instances*, i.e., the users' engagement with the system on a particular task. A new task instance is created whenever a user opens the task interface. A task instance can extend over multiple user sessions and is considered unfinished until the task's output object is submitted to the domain database or the task instance is canceled. When a task instance is completed, the identifier of the created domain output object is recorded in the *OutPKVal* attribute of the TaskInstance table. Thus, each created domain object can be traced back to the particular task instance that produced (or updated) it.

The SessionTaskInstance table of the Logging module captures the chronology of each task instance within user sessions, which may have been performed by multiple users. The system is constrained to allow only one user to work on a task instance at any particular moment.

The detailed key-press level information regarding the users' interactions with input controls within the task instance is also recorded in the Logging module. Whenever a task page is rendered on the screen, the system creates a new record in the EntryField table for each input control on the page. An entry field is an instantiation of an input control within a particular user session. Each entry field is attributable to a single task instance. Users' interactions with entry fields are recorded in the UserEntry table. Every time an input control gains focus, the system records the timing of that event as well as the time of the subsequent loss of focus. It also records the content of the entry field at the time focus was gained, the input entered by the user, and the resulting content of the field at the time the focus was lost.

Taken together, the information contained within these two layers of the Logging module enables a quick and complete reconstruction of a sequence of events as they occurred over time. Importantly, the broader task and user context of these events can also be deduced by exploiting the knowledge of the system and the users embedded within the TIL modules.

Empirical illustrations. In this section, we present three views illustrating the type of information that can be obtained from the usage logs via querying. These illustrations are based on data from an empirical simulation of our prototype, which we conducted for the purpose of collecting usage data. In this simulation, 15 users employed our TIL-enabled prototype to complete several prototypical ERP tasks during a course of over 27 days. The logged data included 39 user sessions and 450 different task instances.

Example 1. Table 1 shows an excerpt from the list of task instances in chronological order by their start time, obtained by querying the TIL model data.

The first five columns in this figure describe the user who initiated the task instance, the id of the task instance (TIID), the start and end time, which refer to the time the task was initiated and completed, and the task name. Also included in the view is the output object for each task instance, specified in

Table 1. Task instance logging.

User	TIID	StartTime	EndTime	TaskName	DTableOut	OutPK Value	Parent TIID
				...			
user3	110	13-Dec 12:17:31	13-Dec 12:23:48	selectTask			
	111	13-Dec 12:17:53	20-Dec 15:07:30	Add PR	Puchase_Requisition	12	
	112	13-Dec 12:19:32	13-Dec 12:22:12	Add PO	Puchase_Order	10	
	113	13-Dec 12:22:24	13-Dec 12:23:39	Add PO	Puchase_Order	11	
user4	115	13-Dec 13:13:08	13-Dec 13:30:27	selectTask			
	116	13-Dec 13:13:35	13-Dec 13:17:38	Add PO	Purchase_Order	12	
	117	13-Dec 13:15:13	13-Dec 13:15:34	Review PR			
				...			
user3	194	20-Dec 15:04:51	20-Dec 15:04:51	AddMaterial	Material	70	
	195	20-Dec 15:05:07	20-Dec 15:06:38	Add GR	Goods_Receipt	6	
	196	20-Dec 15:06:46	20-Dec 15:07:30	Eidt PR	Purchase_Requisition	12	111
	197	20-Dec 15:07:34	20-Dec 15:08:25	Eidt PR	Purchase_Requisition	16	126

columns *DTableOut* and *OutPKValue*. For example, the outcome of task instance 111 is a record in the Purchase_Requisition table with the primary key value of 12. The last column, *ParentTIID*, is filled only for editing tasks, as it connects the instance of an editing task to the task instance that first produced the edited object, as shown for task instances 196 and 197.

The excerpt from the log data demonstrates some of the complexities involved in aggregating this data into demonstration scripts:

– An example of a task instance that extended over more than one user session is presented by task instance 111. This Add Purchase Requisition task started on Dec 13 and ended seven days later, when Edit Purchase Requisition number 196 was completed as the result of user3 submitting the requisition. Importantly, our usage log captures the relationship between Add and Edit task instances, enabling the reconstruction of the process of creating a Purchase Requisition from its inception to its submission.
– Parallel execution of multiple task instances by a single user is demonstrated by the timing of task instances 116 and 117, performed by user4. In determining which tasks are related to each other in a business process, the TIL model relies on the object flow between the tasks. The object flow information enables our playback procedure to determine automatically if two co-occurring task instances are related to each other, and, thus, must be shown together in a task tutorial.
  In the case of task instance 116 (Add Purchase Order) and 117 (Review Purchase Requisition), additional querying that is not shown in Table 1 can reveal if task instance 117 involved reviewing a purchase requisition that was used as input to task instance 116. Based on that information, our playback algorithm would either include or leave out task instance 117 as a part of the playback of task instance 116.

Thus, the task instance data captured by our TIL-enabled prototype contains the complete chronology of the task instances, including the precise determination

Table 2. Interface-level details of user actions within a specific task instance derived from the log.

	Task Instance 111, Add Purchase Requisition, by user3						
	User Entry Time	EFID	ICName	PageNum	Value Start	Entered	Value End
1	13-Dec 12:17:53	1393	Vendor	1		2	2
2	13-Dec 12:18:08	1394	Plant	1		3	3
3	13-Dec 12:18:20	1396	Storage Location	1		2	2
4	13-Dec 12:18:32	1401	Standard Form Button	1	Submit	<press>Submit	Submit
5	13-Dec 12:18:36	1404	Material	2		<press>Lookup	
6	13-Dec 12:18:41	1404	Material	2		<press>Select	
7	13-Dec 12:18:41	1404	Material	2		<receiveSelection>	22
8	13-Dec 12:18:42	1405	Quantity	2	1	2	2
9	13-Dec 12:18:48	1416	Standard Form Button	2	Add Line	<Pres>Add Line	Add Line
10	13-Dec 12:18:48	1417	Material	2		<press>Lookup	
11	13-Dec 12:18:54	1417	Material	2		<press>Select	
12	13-Dec 12:18:54	1417	Material	2		<receiveSelection>	64
13	13-Dec 12:18:55	1418	Quantity	2	1	5	5
14	13-Dec 12:19:05	1414	Form Button	2	Save	<press>Save	Save

of the task instance boundaries in the presence of co-occurring tasks, which is critical for creating task instance visualizations based on usage log data. The logging data also associates each task instance with the resulting changes in the domain data. This allows us to reason effectively about the relationships between the task instances based on the flow of the domain objects.

Example 2. Table 2 details the system-user interactions occurring within a particular task instance. This view aggregates the data from the Logging module so we can trace what happened. It is a chronologically ordered set of all logged user interaction events that occurred while user3 was working on an Add Purchase Requisition task (task instance 111).

As shown in rows 1 through 4, the user entered values in Vendor, Plant, and Storage Location fields on page number 1 of the task interface using the keyboard and then pressed the Submit button. The user subsequently entered the second page of the task interface, where she entered values into fields and saved the resulting purchase requisition for further editing.

In addition to obtaining the details of the input events, we can determine the full set of actual domain input parameters for each task instance by computing the final values in the entry fields of each task page. This is demonstrated by Table 3 for task instance 111. Note that the Quantity values entered for task instance 111 are deliberately left out of Table 3, because quantity is a *non-domain* value, i.e., a number that is not associated with any domain database record. The task instance input and output information plays the key role in determining if task instances are related to each other and in reconstructing process instances for presenting to the user, as described next.

Table 3. Domain objects used as input to task instance 111.

User	TIID	DTable	PKValue Entered
user3	111	Material	22
		Material	64
		Plant	3
		Storage_loc	2
		Vendor	2

3.2 From Individual Tasks to Processes

The flow of business objects between tasks forms a natural basis for our representation of processes. In the course of system use, objects produced as the output of some tasks are used as inputs to others. We therefore define a process in our framework as consisting of a set of tasks related via their inputs and outputs and associated with the type of object that is ultimately produced.

Our system helps the user in identifying those tasks that are relevant to a process, but the user makes the final selection of which tasks comprise a process at the system configuration stage. This specification is stored in the Task module, while the business object that is produced by the process is stored in the Domain module.

Given a process specification as a set of tasks connected via output/input links, *process instances* can be reconstructed automatically from the Logging module within the TIL model. The algorithm for process instance detection is implemented as an SQL procedure over the TIL and Domain data and is best described as a breadth-first search in the *task instance graph*.

A task instance graph is a directed acyclic graph, in which the nodes correspond to task instances, and an arrow from node a to node b is drawn if the output object of task instance a was used as input to task instance b. Figure 3

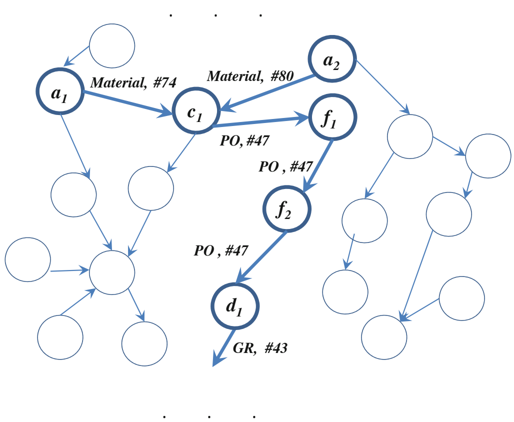


Fig. 3. A fragment of a task instance graph with a process instance highlighted in boldface.

depicts a task instance graph in which one process instance is highlighted in boldface.

The input to the process instance identification algorithm is a specific business object, e.g., good receipt (or GR) #34, and a set of tasks comprising the process, e.g., $P = \{a, b, c, g, e, f\}$, as described in the Task module. The output is a set of task instances comprising the process instance that produced the specified business object. Throughout its operation, the algorithm considers only instances of tasks in the process task set P. It starts by identifying the chronologically latest task instance that produced the specified object as its output and putting that instance in a list of discovered task instances D. The breadth-first traversal then starts from that latest task instance. At each subsequent stage, for each node $\delta \in D$, the search identifies the set of all nodes with links into δ that preceded it in time and adds it to D. The search continues until no new nodes are discovered. The process instance is returned as the set of task instances D.

Note that, due to the optionality of some tasks within a process, there can be great variation in the actual task composition of process instances. Another source of variability is the timing of the task instances involved in a process. Our approach to process instance detection accurately identifies process instances using a SQL-based procedure over the TIL model data, thereby avoiding the pitfalls typically encountered by data mining approaches.

To summarize, the process-related capabilities of our current system prototype include:

Process Configuration Stage. For a given output document type, the user specifies the set of tasks that she wants to have included in the designated process. The tasks must be related to each other via their input-output links [12].

At System Runtime. When a user selects the playback option on a task, one of the choices includes the demonstration of a process involving that task. If the person chooses to play the process, the system queries its log to produce a list of existing completed process instances and presents the choices to the user. The selected instance is demonstrated to the user by the system in the Playback window, as described next.

4 Replaying the Interaction

In this section, we describe our interface and algorithm for creating on-demand playbacks of tasks. We have implemented and tested the interface within our proof-of concept prototype ERP system.

For every available task in the prototype, the user can invoke a ShowMe option, which demonstrates the task interface by playing back a previously recorded system-user interaction of that task. The visualization is performed by the Playback procedure, described in the next subsection, and involves imitation of the recorded user actions and system responses at a speed that makes it easy to follow the process.

Upon invoking the ShowMe option, the user can select demonstrations displaying a task instance that was:

– performed by a particular colleague,
– completed during a specified range of dates, or
– involved a specific business object, e.g., customer invoice number 254.

We chose these parameters because they most closely correspond to the types of information being sought by users in real-life scenarios.

Another playback option is the demonstration of an entire process involving a particular task, as many ERP users in our field studies complained about not understanding where a task fits into an overall process in terms of what tasks preceded it and what tasks followed it. Understanding how tasks are linked and the flow of data through a process is often essential information, particularly when users attempt to address an error caused by a problem with the data, but this information is not readily available from the system.

Figure 4 shows a screenshot of a Playback Parameter Selection window. In this case, the user has specified that they would like to see how user11 has performed the Add Purchase Requisition task within the specified time range. The user selected to view the task within the context of the corresponding Purchase Order process instance.

In response to a ShowMe request, the system queries the Logging module for a list of task or process instances matching that request and presents them to the user. If no parameters are specified, the system generates a list of all available instances of the requested task. At this time, the list of playable task and process

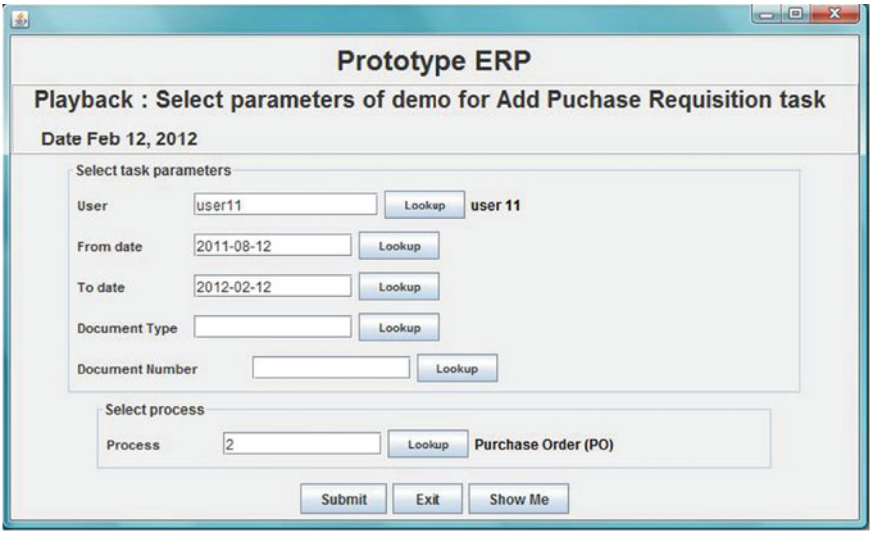


Fig. 4. A screenshot showing the parameters that can be specified for the selection of the task playback.

instances is not rank-ordered in any way. In the future, we plan to investigate effective ways of utilizing the wealth of information available in the logs about the behavior of the user making the request as well as others performing similar tasks for use in rank ordering the choices to best fit the user's profile and goals.

4.1 The Playback Procedure

The pseudocode for the Playback procedure, which is run by the system after the user selects a task instance for demonstration, is depicted in Fig. 5. The procedure is passed a task instance identifier *ti* and the page number within the task interface from which to start the playback. The procedure starts by querying the Task and Interface modules for the composition of the task page, which is required for rendering it (see the call to *getTaskPageInterface* function in line 1). In the next step, the system retrieves the records of user activities for this task instance on this task page from the log in chronological order. We call this set of records a *script*, as it contains the details needed for playing back the task instance. Table 2 depicts the essential columns of the playback script. In step 3, the initial values of the input fields in the task page at the time of

Procedure Playback
Input: task instance *ti*, starting page number *pageNum*
Output: a dynamic visualization of the step-by-step execution of *ti*

```
1  taskPage = getTaskPageInterface(ti, pageNum)
2  script = getTIScriptFromLogs(ti, pageNum)
3  initializeTaskPageInputFields (taskPage, script)
4  display(taskPage)
5  remove records of ineffectual actions from script
6  for each record in script
7        if (inputControlType(record) == INPUT_FIELD )
8                  showExecution(record)
9        else if  (inputControlType(record) == BUTTON)
10                 showExecution(record)
11                 if (getButtonType(record) == SUBMIT_BUTTON and
12                      pageNum != lastPage(ti))
13                      Playback(ti, pageNum + 1)
14                 end if
15       else if  (inputControlType(record) == MENU_ITEM)
16                 relatedTask = taskInvokedViaMenu (record)
17                 relatedTI = findNextTI(task, sessionID, record.StartTime)
18                if (relatedTI != NULL) Playback(relatedTI, 1) endif
19       endif
20 end for each
```

Fig. 5. Pseudocode for procedure Playback.

ti are determined based on the *script*; the page is then displayed in a separate Playback window.

For the effectiveness of the demo, the playback script is pre-processed in step 5 to remove any obviously ineffectual key/mouse press sequences. For example, consider the situation in which a user, having invoked a lookup interface, presses the "Display All" button (shown in Fig. 1) more than once. None of the button presses after the first one lead to a change in the state of the system and are therefore eliminated from the playback. Another example is a canceled lookup action, which would also add no value.

Lines 6–20 of the pseudocode specify a loop in which every record of the *script* is visualized. Recall that each record contains information on the invoked input component. The record is first analyzed to determine what kind of input component is being referenced. In the case where the record refers to interaction with an input field:

- if the value was typed into the field, as specified, for example, by the log record in row 1 in Table 2 for the Vendor value, the *showExecution*(*record*) function displays the keyed value being entered directly into the text field, or
- if the lookup functionality was invoked (see log records displayed in rows 5–7 in Table 2), the *showExecution*(*record*) function shows the interaction occurring within the lookup interface followed by the transfer of the value selected from the list into the text field.

In the case where the record refers to a button press: the *showExecution*(*record*) function visualizes the button press and its visible effects, such as any informational prompts that occur in response to the press event. If the Submit button was pressed, the Playback procedure is called recursively to visualize the interaction on the next page of the task instance (lines 11–13).

Since menu items in the interface provide quick access to tasks that are closely related to the current one, invoking a menu item triggers another call to the Playback procedure. First, however, which task interface was enacted ($relatedTask$ in line 16) must be determined by consulting the Interface and Task modules and, from that information, finding the corresponding task instance ($relatedTI$ in line 17). The $relatedTI$ task instance is identified by the *findNextTI*() function as the one occurring closest in time within the current user session to the invocation of the menu instance. If that task instance did not involve any objects used as the input to the currently visualized task instance *ti*, it is discarded. Otherwise, the identified $relatedTI$ is visualized in a separate window via the call to Playback ($relatedTI$,1) in line 18.

5 Conclusions and Future Work

The application of collaboration theory to creating software interfaces has been explored primarily by researchers in the field of Artificial Intelligence (e.g. [7, 28]). In our research, we have employed collaboration theory to examine the usability bottlenecks of ERP systems and derive design principles for improving the systems' support of user activities [9]. We have presented a tutorial-by-demonstration tool that implements one of our design principles, namely that the system must be able to help its users learn to navigate and operate it effectively.

Our approach illustrates how usage histories collected by the system can be put to use for automatically creating demonstrations of the system interface as dynamic visualizations of previously recorded system-user interactions. We have outlined the Task-Interface-Log (TIL) data model that is used to support such demonstrations and discussed the components that are critical to enabling their efficient, on-demand composition from key-press-level and task-level log data. The prototype we described is implemented in Java and MySQL.

A number of simplifications in our prototype have been made; notably, the number of tasks and variety of input components are more limited than those found in modern ERP systems. Our intention is not to replicate the scope and scale of these systems but rather to demonstrate the feasibility of our approach and test the design of its critical components, i.e., the TIL model and the algorithms for on-demand task and process instance identification and dynamic playback. In the future, we plan to work on user-testing and fine-tuning our initial proof-of-concept version of the playback interface presented here.

Given the complexity of ERP system interfaces, the ability to replay previously occurring interactions is beneficial to users in a variety of circumstances, including new users performing a process for the first time, more experienced users performing particular tasks on an infrequent basis, or any user who just does not remember how to proceed. Building the ability to dynamically generate automated playbacks of user-system interactions right into the system is also a more efficient and lower cost option than having users rely on others within the organization for help.

In our framework, processes are formed by following the logical flow of business objects between tasks, with the system helping the user identify those tasks that are relevant to each process. Defined in this way, the concept of a process is flexible, yet process instances can be effectively and efficiently reconstructed from the logs. The rich data set of past interactions can also be exploited for rank-ordering demonstrations to best fit the user's needs. Additional applications of log data analysis to user support enabled by our representation include providing the user with recommendations regarding the next task to perform in a process, determining the user with the most experience in performing a particular task, computing a list of incomplete tasks for a user, enabling annotations and user-editing of tutorial playback scripts, and auditing the usage history for compliance with process guidelines. We are exploring several of these possibilities in furthering our long-term research goal of making organizational systems function as their users' helpful and intelligent partners.

Acknowledgements. This material is based in part upon work supported by the National Science Foundation under Grant No. 0819333. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Hestermann, C.: Key issues for enterprise resource planning. Gartner (2009)
2. Hamerman, P.: ERP applications 2007: innovation rekindles. Forrester Research (2007)
3. Iansiti, M.: Erp end-user business productivity: a field study of sap & microsoft. Technical report, Keystone Strategy (2007)
4. Topi, H., Lucas, W., Babaian, T.: Identifying usability issues with an ERP implementation. In: Proceedings of the International Conference on, Enterprise Information Systems (ICEIS-2005), pp. 128–133 (2005)
5. Cooprider, J., Topi, H., Xu, J., Dias, M., Babaian, T., Lucas, W.: A collaboration model for erp user-system interaction. In: Proceedings of HICCS'2010, pp. 1–9. IEEE Computer Society (2010)
6. Terveen, L.G.: Overview of human-computer collaboration. Knowl.-Based Syst. **8**(2–3), 67–81 (1995)
7. Grosz, B.G.: Beyond mice and menus. Proc. Am. Philos. Soc. **149**(4), 529–543 (2005)
8. Shieber, S.: A call for collaborative interfaces. ACM Comput. Surv. **28A** (electronic). http://www.acm.org/pubs/citations/journals/surveys/1996-28-4es/a143-shieber/ (1996)
9. Babaian, T., Lucas, W., Xu, J., Topi, H.: Usability through system-user collaboration. In: Winter, R., Zhao, J.L., Aier, S., et al. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 394–409. Springer, Heidelberg (2010)
10. Leshed, G., Haber, E.M., Matthews, T., Lau, T.: Coscripter: automating & sharing how-to knowledge in the enterprise. In: CHI '08: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human factors in Computing Systems, pp. 1719–1728. ACM, New York (2008)
11. Ivory, M.Y., Hearst, M.A.: The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. **33**(4), 470–516 (2001)
12. Lucas, W., Babaian, T.: Implementing design principles for collaborative ERP systems. In: Peffers, K., Rothenberger, M., Kuechler, B., et al. (eds.) DESRIST 2012. LNCS, vol. 7286, pp. 88–107. Springer, Heidelberg (2012)
13. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Inf. Syst. **33**(1), 64–95 (2008)
14. Quast, M., Handel, M.J.: Social information systems - the end of shadow applications? In: Cordeiro, J., Maciaszek, L.A., Filipe, J. (eds.) ICEIS 2012. LNBIP, vol. 141, pp. 5–15. Springer, Heidelberg (2012)
15. Plaisant, C., Shneiderman, B.: Show me! guidelines for producing recorded demonstrations. In: Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 171–178. IEEE Computer Society (2005)
16. Grossman, T., Fitzmaurice, G.: Toolclips: an investigation of contextual video assistance for functionality understanding. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI '10, pp. 1515–1524. ACM (2010)
17. Chakravarthi, Y.A., Lutteroth, C., Weber, G.: Aimhelp: generating help for gui applications automatically. In: Proceedings of the 10th International Conference NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction. CHINZ '09, pp. 21–28. ACM, New York (2009)
18. Caldwell, D.E., White, M.: Cogenthelp: a tool for authoring dynamically generated help for java guis. In: Proceedings of the 15th Annual International Conference on Computer Documentation, SIGDOC '97, pp. 17–22. ACM, New York (1997)

19. Yeh, T., Chang, T.H., Xie, B., Walsh, G., Watkins, I., Wongsuphasawat, K., Huang, M., Davis, L.S., Bederson, B.B.: Creating contextual help for guis using screenshots. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11, pp. 145–154. ACM, New York (2011)
20. Ramachandran, A., Young, R.M.: Providing intelligent help across applications in dynamic user and environment contexts. In: Proceedings of the 10th International Conference on Intelligent User Interfaces, IUI '05, pp. 269–271. ACM, New York (2005)
21. Brusilovsky, P., Cooper, D.W.: Domain, task, and user models for an adaptive hypermedia performance support system. In: IUI '02: Proceedings of the 7th International Conference on Intelligent User Interfaces, pp. 23–30. ACM Press, New York (2002)
22. Linton, F., Joy, D., Schaefer, H.P., Charron, A.: Owl: a recommender system for organization-wide learning. Educ. Technol. Soc. **3**(1), 62–76 (2000)
23. Shen, J., Fitzhenry, E., Dietterich, T.G.: Discovering frequent work procedures from resource connections. In: Proceedings of the 13th International Conference on Intelligent User, Interfaces, pp. 277–285 (2009)
24. van der Aalst, W.M.P., Pesic, M., Song, M.: Beyond process mining: from the past to present and future. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 38–52. Springer, Heidelberg (2010)
25. Dorn, C., Burkhart, T., Werth, D., Dustdar, S.: Self-adjusting recommendations for people-driven ad-hoc processes. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 327–342. Springer, Heidelberg (2010)
26. Greco, G., Guzzo, A., Sacc, D.: Mining and reasoning on workflows. IEEE Trans. Knowl. Data Eng. **17**, 519–534 (2005)
27. van der Aalst, W.M.P.: Process discovery: capturing the invisible. IEEE Comp. Int. Mag. **5**(1), 28–41 (2010)
28. Rich, C., Sidner, C.L., Lesh, N.: Collagen: applying collaborative discourse theory to human-computer interaction. AI Mag. **22**(4), 15–25 (2001)

Measuring Users' Emotions with a Component-Based Approach

Rogério Aparecido Campanari Xavier(✉) and Vânia Paula de Almeida Neris

Sustainable and Flexible Interaction Laboratory, Department of Computing, Federal University of São Carlos, Rod Washington Luiz, Km 235, São Carlos, Brazil
{rogerioacx, vania}@dc.ufscar.br

Abstract. The emotional experience is present in all types of interaction between human beings. Therefore, it is vital to consider the emotional experience during the design process of users' interfaces to inspire greater confidence among users as well as for the interface to be easier to learn and use. Considering the complexity of the subject, it is necessary to combine methods to minimize the detection of false positives in the evaluation of the user's emotional state while interacting with information systems. This work presents a hybrid evaluation approach based on the emotion model described by Scherer [1]. The proposed approach allows designers to check whether the information system creates a positive, neutral or negative emotional reaction in the users. A feasibility study was conducted in which an emotional evaluation of a web system was performed based on a group of elderly users using tablet devices.

Keywords: Emotion · Users' emotional experience · Emotional evaluation · Information systems

1 Introduction

Emotions are a key to understanding human behavior [2]. They are seen as a mental state that arises spontaneously, without conscious effort, and considered as feelings in general that are often accompanied by physiological changes such as breathing, circulation and secretions. They are also influenced by several external and internal stimuli including the context of the situation, life experience, recent experiences, personality, affect and the cognitive interpretation of these influences [3].

Emotions are a complex phenomenon that are difficult to identify and to measure. For psychologists, emotions are linked to the reaction of several components as cognitive aspects, subjective feelings, behavioral tendencies, physiological responses and motor expressions [4, 5]. Emotions affect our attention, perception, memory, behavior and cognition [6]. Emotional responses are present in all types of interaction between human beings. The affective system lead us to quickly determine if the elements of the environment we live in are safe or dangerous, or good or bad; [7, 8]. This type of knowledge about affective systems may explain why people express their feelings and emotional responses while interacting with information systems.

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 393–409, 2013.
DOI: 10.1007/978-3-642-40654-6_24,

Therefore, it is vital to consider the users' emotional experience during the design process to allow the information system's interfaces to inspire greater confidence among users as well as for the interface to be easier to learn and use [9]. In the Human–Computer Interaction (HCI) literature, it is possible to identify methods, techniques and tools for the assessment of users' emotions, (e.g., [5, 10–12]). However, the separate applications of these approaches may lead to the detection of false positives in the identification of the users' emotional state due the interaction process.

Among elderly users, we observed that in some cases, a bad interaction experience relative to the traditional usability metrics, such as the time of interaction, the number of mistakes or non-concluded tasks, whereas in other cases, the users selected symbols indicating a good emotional experience, such as happiness or satisfaction in a self-reported questionnaire. One user stated, "*During life, we cannot be sad*" and selected a happy face in the questionnaire. Although we agree that usability metrics are not by themselves sufficient for one to judge an interaction experience, in this case, the user considered her life experience in general and did not evaluate the emotional reaction due to her interaction with the system. We argue that it is necessary to combine methods, involving different stakeholders such as users and specialists, to minimize the detection of false positives.

This work presents a hybrid evaluation approach that allows designers to check whether the information system creates a positive, neutral or negative emotional reaction in the users. Our approach, which is based on the emotion model described by Scherer [1], considers a set of methods that allows designers to identify the emotional state of the user by considering the subjective feelings and physiological reactions with the user's opinions and reactions as well as the cognitive appraisals, behavioral tendencies and motor expressions. A feasibility study was conducted in which an emotional evaluation of a web system was performed that considered a group of elderly users using tablet devices.

This paper is organized as follows. Section 2 presents some models found in the literature for describing emotions. Section 3 summarizes some techniques and tools that can be applied to evaluate each component of the model proposed by Scherer. Section 4 describes our hybrid approach aiming to help designers in evaluating the user's emotional response. Section 5 presents the feasibility study performed to evaluate the proposed approach. Section 6 presents a critical analysis and some of the lessons learned. Finally, Sect. 7 concludes.

2 Models for Emotions

The identification of human emotional states is difficult and complex [13]. Therefore, to try to gain a better understanding of the subject, some models describing how we feel emotions can be found in literature. Some of these models describe the emotions using mainly a cognitive approach (e.g., [14]), whereas others consider multidimensional aspects as pleasure and arousal (e.g., [15, 16]). It is also possible to find models that describe the emotions related to specific contexts; emotions elicited by products, in a process-level approach [7] or in an appraisal approach [17]. There are also those that describe emotions through a component-based approach [1]. A brief summary of each of these models and the choice of the model adopted in this work are presented following.

2.1 Cognitive Model

The model proposed by Ortony, Clore and Collins [14] is based on the cognitive theory of emotions. This model is classified into three broad classes according to the cognitive focus involved. The focus is related to three aspects or possible situations which are: *consequences of events*, *actions of agents* or *aspects of objects*. Each of these three aspects is related to an affective reaction. By focusing on events, it is possible to get an emotional reaction in terms of satisfaction or dissatisfaction. Each event can provide consequence to the individual himself or someone else.

The focus on agents leads to affective reactions related to the approval or disapproval of a specific action done by the individual himself or by the action of another person. The focus on objects is mostly based on liking or disliking the attributes of the objects. In this case, affective reactions, positive or negative, are generated from the perception of how good or bad the situation is which takes into account the point of view of the individual.

2.2 Dimensional Model

The model developed by Russell [18] was originally created to assist psychologists to represent the structure of emotional experiences. This model is also known as the circumplex model of affect or core affect. Briefly, it consists of a circle-shaped structure, which classifies 28 emotions in adjectives classified in two main dimensions: pleasure-displeasure and excitement-calm. Thus, each emotion can be seen as a point in common between these two dimensions, or as the degree of both dimensions of pleasure and arousal.

2.3 Process-Level Model

The model developed by Norman [7] acts on different levels of information processing which helps to explain the relationship between the product and the users' emotional aspects. This model considers is classified in three levels: *visceral, behavioral and reflective*.

The "*visceral level*" is responsible for evaluating automatic reactions to external stimuli about what is good or bad, safe or dangerous. It is responsible for making an immediate analysis and communicating the individual's motor system. This level does not allow reasoning. It is concerned about the appearance and potential immediate attraction of the product. The visceral level is related to the immediate visual influence that the product reflects on the user, causing positive or negative reactions. Moreover, it can be influenced by the other levels.

The "*behavioral level*" consists of processes that control the behavior, pleasure and effectiveness of use (function and operability) elicited by product. This level is connected to the qualities of effectiveness and functional aspects of the product and the ease of understanding and operating the product. The behavioral level can be influenced by the reflexive level and can influence the visceral level confirming or not the first impression about the product generated by the visceral level.

The third and last level is called "*reflective level*". It is related to conscious thought, learning new concepts and the self-image of the user. This level considers the rationalization and intellectualization related to the product (the story that the product "*tells*"). It also involves affective memories. It is linked to individual and cultural meanings that the product represents for people on a subjective point of view, the personal satisfaction built on long-term, such as the fidelity to a particular brand. The reflective level does not have direct access to sensorial information. It can only influence the behavioral level.

2.4 Appraisal Model

This model is based on the cognitive appraisals theory proposed by Desmet [17]. It is also known as a multilayered model of product emotions. This basic model considers that there are four parameters responsible for the generation of an emotion, they are: *appraisal*, *concern*, *product* and *emotion*. The *concern* and *product* parameters are considered requirements for the *appraisal* parameter. The *appraisal* is responsible for determining whether the product is able to provoke an emotion or not, and if so, which emotion is that.

The *concern* parameter is responsible for relating the user's interests for the product. These concerns are used as preferences in the appraisal process. Products that meet the interests of the users are evaluated as beneficial and therefore will cause positive emotions. The opposite is rated as harmful and contributes to the generation of negative emotions.

The *product* parameter is related to stimuli (feelings and fantasy) that the user attaches to the product, responsible up provoking emotions. A stimulus can be a real event as an advertisement of shoes on the internet. Moreover, a stimulus can also come from an event remembered or imagined, i.e., the desire of a person to win the lottery. In summary, emotions can be related to material artifact, or a representation of something.

The fourth and final parameter identifies the emotion generated by the relationship between the person and the product. This emotion considers the appraisal made by the person in relation to the stimuli that the product offers and the individual personal concerns.

2.5 Component-Based Model

There is no right or wrong model to describe the emotions. Each model has a specific focus and the choice for one of them should take into account the designers' expectations and needs. The cognitive model proposed by Ortony, Clore and Collins [14], for instance, has been used, in the Affective Computing area, as a model for synthesize emotions and affects, and its main idea has been implemented in computers [20].

The multidimensional model is primarily focused on the user subjective experience. As subjective feelings are restricted to the conscious experience of emotions, the identification of emotional states is commonly accessed through self-report instruments such as questionnaires [21].

The process-level model, developed by Norman, explains how humans relate emotion with objects cognitively. Emotions are considered add-ons to cognition. This model considers the isolated experience in a single person's head, not considering, for example, the social and interactive experience [22].

The appraisals model [17] helps designers to explain how different emotions are elicited by different underlying appraisals. This model is more abstract than the model-level process, making it more difficult to be applied in a practical design experience [23].

In this work, we have adopted Scherer's model [1], which is based on components. According to him, "*it is interesting to speculate about the possibility that specific components of emotion are specialized to serve specific functions*" (p. 297). Scherer's model was chosen because its approach based on components allows us to work with each component separately and therefore choose the appropriate methods to evaluate the different dimensions. Moreover, it has been successfully used in other HCI studies to support the investigation of emotional experiences in interactive contexts (e.g., [4, 24, 25]).

Scherer's model consists of a triangle, which is connected to two components: cognitive appraisals and behavioral tendencies [1, 4, 5]. The cognitive appraisals are relevant to the assessment of the environment including the objects and events. This component leads to different emotions depending on the user's interpretation. In contrast, the behavioral tendencies prepare the user's emotional reactions. According

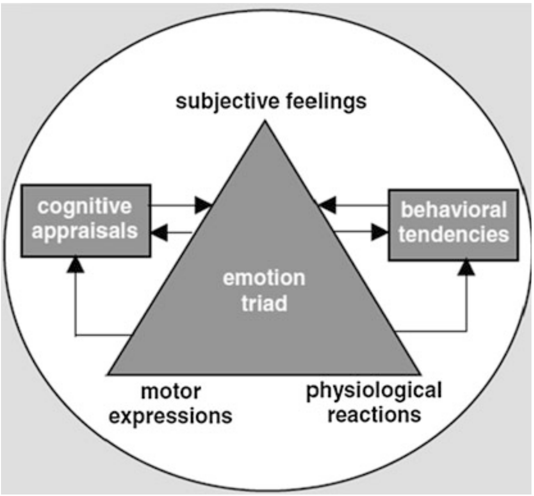


Fig. 1. Scherer's model to describe emotions (extracted from [4]).

to Mahlke and Mingue [4], these reactions can be expressed in several ways, such as the time required for single input operations or completing a defined goal, the accuracy of reaching a goal, the number of errors and the number of creative ideas during interaction with a system. Figure 1 illustrates Scherer's model.

In addition to the cognitive appraisals and behavioral tendencies, Scherer's model also considers the following:

- Subjective Feelings that monitor the internal state and the organism's interaction with the environment, also known as conscience of emotional state;
- Motor expressions that communicate reactions and emotional and behavioral tendencies; and
- Physiological reactions that act to regulate the system, determining the activation of neuroendocrine processes (related to the nervous and endocrine influences) such as heart rate, skin conductance, blood pressure, respiration, and pupil dilation [19].

Our approach considers a set of methods that allows designers to identify the user's subjective feelings as reported by the user as well as cognitive appraisals and motor expressions derived from the participation of evaluators.

3 Emotional Evaluation

In the literature, it is possible to identify methods, techniques and tools with which to assess emotions in humans. Each of them has features characterizing them more appropriate for certain aspects of emotions. Generally, the instruments applied in the methods, techniques and tools can be classified as verbal or non-verbal. In this research, we consider an instrument as verbal when the user explicitly verbalizes what s/he is feeling.

According to Desmet [25], verbal instruments enable users to express their emotion in scales (when a user says "*I am very happy*" or "*I am not anxious at all*") and to report "*mixed*" emotions as tension. However, they are difficult to apply across cultures because it may not be easy to translate emotions into words. On the other hand, non-verbal instruments can be considered discreet and independent of culture and language. However, they can be subjective as they generally use universal symbols such as pictograms.

Figure 2 presents a taxonomy, which classifies emotional assessment metrics as verbal or non-verbal. Each final node in the taxonomy represents one of the five components of Scherer's model, and each parent node represents a set of available methods, techniques, tools and instruments that can be used to measure a component.

Cognitive assessments are linked to the interpretation of a situation and further development of emotions. This component can be measured by the Geneva Appraisal Questionnaire (GAF) [26], the Think-aloud method [27] and the Subjective Discourse Analysis [28]. Although the Subjective Discourse Analysis considers spoken statements, this technique was classified as non-verbal because users do not explicitly say what they are feeling. The evaluators should interpret the statements spoken during the user's interaction and classify the related emotion.

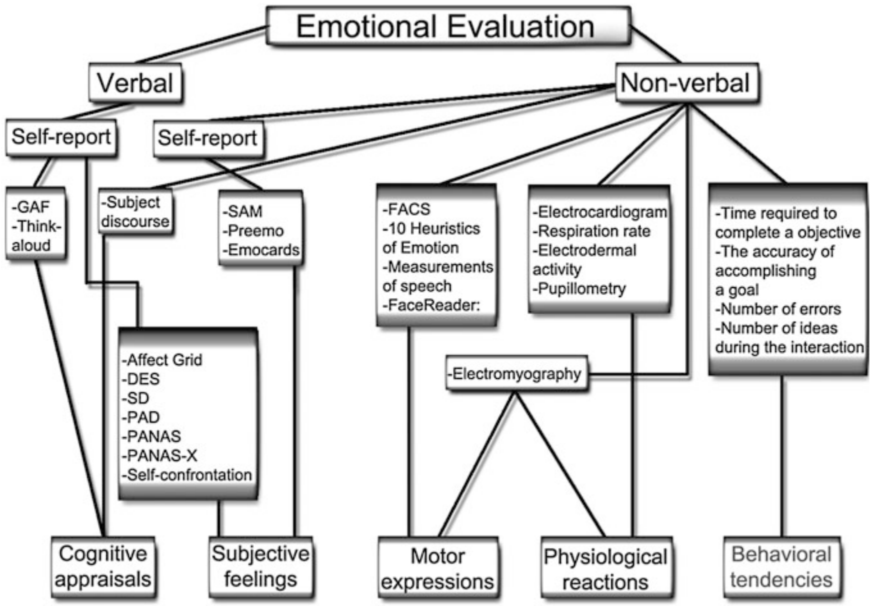


Fig. 2. A taxonomy for emotional evaluation methods, techniques and tools.

According to Scherer [5] and Desmet [25], there is no objective method capable of measuring subjective feelings. It is necessary to query the user, and thus, the methods involve self-assessment. Most of the methods for evaluating subjective feelings are non-verbal, such as the SAM (Self-Assessment Manikin) [29], Emocards [30] and Preemo [25]. Verbal instruments can also be found in the literature, as the Affect Grid [11], Differential Emotions Scale (DES) [31], Semantic Differential (SD) [32], Pleasure, Arousal, Dominance (PAD) [33], Positive and Negative Affect Schedule (PANAS) [34], Positive and Negative Affect Schedule – Expanded (PANAS-X) [35] and Self-confrontation [36].

The motor expressions are related to facial movements, body gestures as well as to some characteristics of speech as speed, intensity, melody and sound. Methods that can be applied include the Facial Action Coding System (FACS) [37], the Ten Heuristics of Emotion [10] and electromyography. A tool that can be used for automatic analysis of facial expressions is the FaceReader [38].

Physiological reactions are non-verbal and can be measured by electrocardiogram, respiration rate, electrodermal activity, electromyography, pupillometry, etc. Physiological reactions allow designers to evaluate the user's emotional responses in an experimental context once the users spontaneously and unconsciously reveal their emotions [12]. However, most of these evaluations require expensive instruments and are intrusive and complex [39]. A study concerning the measurement of emotions based on different physiological parameters can be found in [40].

Finally, behavioral tendencies are also non- verbal and generally are evaluated by performance indicators, such as the time required to complete a task, the accuracy of reaching a goal, the number of errors and the number of creative' ideas during the interaction [4].

4 A Hybrid Approach

Aiming to minimize the detection of false positives in the emotional evaluation of information systems, this work proposes a hybrid approach based on the emotion model described by Scherer [1]. Considering the evaluation methods and instruments presented in the literature, we have selected a subset that matches the five components of the Scherer's model. This selection considered the main stakeholder (user or specialist) responsible for the final result of each method or instrument, aiming to balance the final emotional assessment of the information system.

The subjective feelings demand a self-report instrument, and we have selected the SAM [29] to measure it. The SAM is composed of three sets of figures that represent pleasure, arousal and dominance. Each dimension is evaluated using a scale ranging from 1-9 in which the user selects one circle. As a pictogram, it reduces the cultural differences, and as a non-verbal instrument, it avoids problems with the verbalization of emotions. Moreover, it considers the dominance aspect explicitly.

In our approach, we suggest that SAM be applied after the interaction with the system. The evaluator presents the instrument to the user and asks him/her to classify his/her experience by choosing one of the nine circles in each dimension (pleasure, arousal and dominance). Figure 3 illustrates the pictograms adopted by SAM for the pleasure dimension. The emotional response is considered positive if the user selects one of the circles indicated by the V+. The negative experience options are represented by V–, and the neutral experience is the central option and represented by VN. The final result can be reached for each dimension by adding the number of votes of all users in V+, VN and V−.

Following Mahlke and Mingue [4], we have adopted effective and efficient metrics to measure the behavioral tendencies. They include the time required to complete a task and the number of errors or help requests, among others. Designers may not have difficulty collecting these metrics and evaluating if the final result is positive, negative or neutral, as they are commonly measured in traditional usability tests.

The physiological reactions can be assessed by sensors because they are related to neuroendocrine processes. The data should be collected during the interaction instead of only at the end of interaction. Moreover, the sensor should not disturb the user during the interaction. The results should be compared to baseline values established

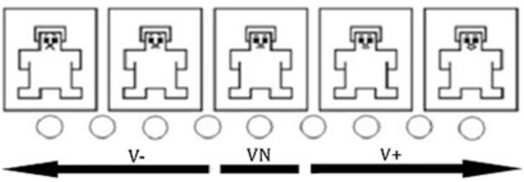


Fig. 3. Pictograms adopted by the SAM questionnaire for measuring pleasure and the emotional values scales.

by the designers. If heart rate is collected, a baseline value could be 85 beats per minute. With a baseline value, designers can evaluate if the final result is positive, negative or neutral. However, sensors that can be used during the interaction, which save the data and do not disturb the user, are generally expensive.

The motor expressions component is assessed in our approach by the Ten Emotion Heuristics [10], which are frowning, raising eyebrows, looking from a distance, smiling, compressing the lips, moving his mouth, vocal expressions, hand touching the face, going back to the chair and leaning the trunk forward. The evaluation is divided into two steps. In the first one, a group of pre-selected appraisers watch videos of the user's interaction. The videos can be captured by common webcams and should record the user's face and body.

For each heuristic identified, we recommend that the appraisers register the time it occurred, the task the user was doing, the heuristic or set of heuristics identified and a description of the emotional aspects. In the second step, the evaluators meet, discuss and build a final list containing the heuristics found. According to Lera and Domingo [10], the final emotional experience evaluation is set as negative if five or more negative heuristics are found per user.

Because it is a heuristic evaluation, the collective common sense can identify the emotional experience of a group of users more accurately than an individual appraiser. Furthermore, facial recognition software is generally expensive, and FACS demands an experienced assessor to apply it properly.

The cognitive appraisals are measured by an adaptation of the Subjective Discourse Analysis [28]. Using the same video as that recorded for the heuristic evaluation, the assessor lists key-expressions that were spoken spontaneously by the users during the interaction. Key expressions are central ideas that represent a synthesis of the discursive content, such as "*And now?*" or "*Should I click here?*". In addition to the expression, the assessor should add a description of the emotional situation in which that phrase was said, for example, in a moment of confusion, joy, or surprise. Thus, based on the description, the evaluator classifies the expression in terms of positive, neutral or negative.

After analyzing the videos and selecting the key expressions for each user, a final list of key expressions should be created. To be selected for the final list, a key expression should be used by more than one user. Finally, considering the most frequently spoken expressions and their classification as positive, negative or neutral, the designer can define the users' emotional response to that system considering the cognitive appraisals. Figure 4 shows the hybrid approach for assessing the emotional state of the users.

Applying the methods as described here, the designer has partial and complementary emotional responses of the users to the interaction with the information system. Considering the evaluation of each component, it is possible to reach a more comprehensive result and to decide if the information system creates a positive, neutral or negative emotional reaction in the users.

Moreover, the results from each component support designers altering different aspects of the interface. Considering the most recurrent heuristic, for instance, it is possible to learn if the users are frustrated or confused. By SAM, it is possible to see if the users are excited but not confident, and by the sensor results, it is possible to see if the users are anxious. Finally, Table 1 presents the methods that are part of our

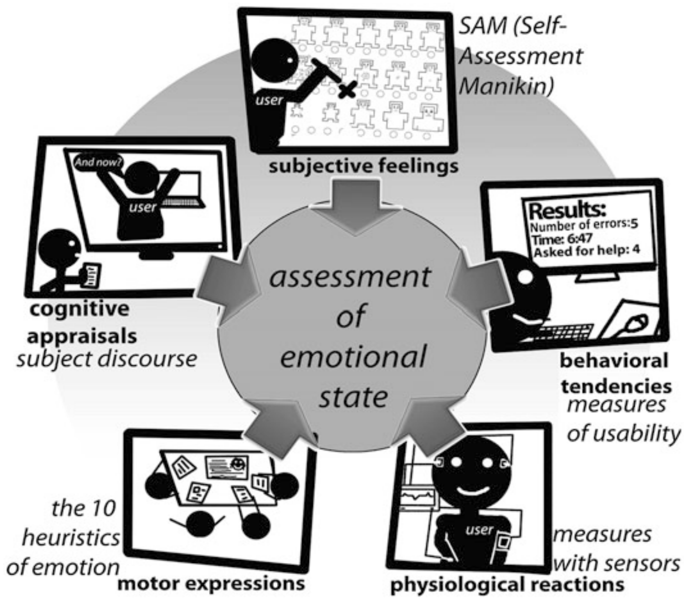


Fig. 4. A hybrid approach for assessing the emotional state of users.

Table 1. Stakeholders and the final decision on the user's emotional response.

Scherer's component	Method	Stakeholder
Subjective feelings	SAM	User
Behavioral tendencies	Effective and efficient metrics	Designer
Physiological reactions	Sensors	User
Motor expressions	Ten Heuristics of Emotion	Group of designers
Cognitive appraisals	Subjective Discourse Analysis	Designer

approach and the stakeholder who makes the final decision regarding the user's emotional response.

We argue that these complementary views minimize the detection of false positives because we consider not only the information provided by users but also the designers' opinions to classify the emotional experiences.

The next section presents a feasibility study applying the proposed approach.

5 Feasibility Study

Aiming to assess the feasibility of our hybrid approach, we selected a group of seven elderly users to evaluate a website about food and recipes using tablet devices. The evaluation occurred in the Social Reference Assistance Center (CRAS, in its Portuguese acronym) in São Carlos-SP, Brazil. The users were selected according to their

Table 2. Characteristics of the elderly users.

User	Genre	Age	Level of Education	Experience in computing
1	Feminine	60	Elementary school	No experience
2	Feminine	71	No schooling	No experience
3	Feminine	57	Fundamental school	No experience
4	Feminine	66	Elementary school	No experience
5	Feminine	66	Elementary school	No experience
6	Feminine	61	High school	Between 1 and 2 years
7	Feminine	58	Elementary school	No experience

Table 3. Results of the SAM assessment.

Pleasure			Arousal			Dominance		
V-	VN	V+	V–	VN	V+	V–	VN	V+
0	0	7	0	1	6	6	0	1

age, education level and experience with devices and were asked to find a specific recipe, starting from the website home page. Table 2 summarizes the users' characteristics. Elementary school in Brazil considers the first four years of studies. Fundamental school considers the first eight years of studies. As the elderly users were not accustomed to using to tablet devices, one of the researchers acted as an active moderator, answering questions when the users asked for help.

Each user was filmed by two common cameras. One camera was focused on the user's body, and the other was registering the user's interaction with the device. After each interaction, the users were asked to fill in the SAM questionnaire. The users' selections were added in each dimension, and Table 3 summarizes the results. The pleasure dimension had a positive assessment with seven positive votes (V+). The arousal dimension was also ranked as positive with six positive votes (V+). On the other hand, the dominance dimension had a negative assessment with six negative votes (V–). The users reported that they had a pleasurable and exciting interaction but that they were not in the control of it.

The evaluation was performed by a group of six evaluators, including five from the computer science field and a professional nurse. The nurse was invited to join the group based on the idea that a professional from the health area could provide a complementary view in an evaluation with elderly users. One of the evaluators had experience in applying the method, and the others received one-hour of training. The final video, with all of the users, was 53 min and 41 s long. Figure 5 illustrates three different moments in the video and the heuristics identified.

After the interaction experience, we applied the Ten Heuristics of Emotion method.

In this feasibility study, three users had a positive experience with less than three different negative heuristics identified, and four had a negative experience with five or more negative heuristics identified. With a small number of users and a non-

Fig. 5. Three examples of emotional heuristics.

expressive difference in the final result, we classified the assessment of motor expressions as neutral.

The Subjective Discourse Analysis, as described in our approach, was applied by the researcher that acted as the moderator during the interaction. Analyzing the same video used to evaluate the heuristics, it was possible to identify the key statements made by the users in addition to their interaction context. The selected statements were classified as positive, neutral or negative. Furthermore, the most frequently spoken key statements were considered in the final cognitive appraisals evaluation. Table 4 summarizes the data collected.

Checking the list of the top five spoken expressions, i.e., those most frequently stated by the users, we evaluated the cognitive appraisal as negative. The top five statements were chosen, and these statements were sufficient to indicate an emotional response tendency. If a non-clear tendency is reached in a study, designers should consider additional frequently spoken statements.

The behavioral tendencies were evaluated considering the number of times the users asked for help while interacting. As the expected interaction time was short (approximately two minutes), we considered that up to two solicitations for help

Table 4. The most spoken key expressions.

Key expressions	Interaction context	Experience	Number of users
Here?	Demonstrate doubt or do not know how to proceed in the task	Negative	5
Recipes!	Realize where to click to accomplish the task	Positive	4
And now?	Perform an action and do not know what will happen. It was also said in moments of frustration or when the user was confused	Negative	4
I do not know	Difficulty in understanding how the system works, how to interact, or when abandoning the activity	Negative	3
I do not understand	Feelings of confusion or disorientation due to do not knowing how the information was presented in the interface	Negative	3

Table 5. Results of the emotional assessment to the cooking site using the hybrid approach.

Scherer's component	Method	Evaluation
Subjective feelings	SAM- dimension of pleasure	Positive
	SAM- dimension of arousal	Positive
	SAM- dimension of dominance	Negative
Behavioral tendencies	Effective and efficient metrics	Negative
Physiological reactions	Sensors	Not applied
Motor expressions	Ten Heuristics of Emotion	Neutral
Cognitive appraisals	Subjective Discourse Analysis	Negative

would be classified as positive. Three solicitations for help were classified as neutral, and more than 3 solicitations were classified as negative. Two users had a positive behavioral tendency, and one user's tendency was classified as neutral. Four users asked for help more than three times, and their behavioral tendencies were classified as negative. The final result for this component was deemed negative.

The physiological measures were not evaluated in this feasibility study because of the high cost of the specific equipment and sensors required. Furthermore, this type of assessment could be intrusive and influence the emotional responses of users. Table 5 presents the final results of the website emotional evaluation.

Thus, an analysis of the results of each evaluation revealed that the emotional state of users while interacting with the website was classified as negative.

6 Critical Analysis and Lessons Learned

Using this hybrid approach, we were able to identify a set of relevant information about the emotional experience of users. For instance, applying the SAM questionnaire, it was possible to realize that even when elderly users do not have control over technology, the interaction can be pleasurable and excited. Other lessons learned include the following:

- Even in an evaluation of a user group with similar profile characteristics, there were variations in the users' emotional states.
- The users who were not familiar with the menus and search engines triggered a greater number of negative heuristics. This finding suggests a bad relationship between less experience and the emotional response, i.e., the less the user knows about the system interaction logic, the more negative is the emotional response to the interaction.
- The use of the Subjective Discourse Analysis allowed us to observe that some key statements are made by most of the users in similar interaction experiences. This observation suggests that affective systems could recognize these statements and change their user interfaces based on their occurrences.
- The evaluators who applied the Ten Heuristics of Emotion noted that the process of evaluating the heuristics requires a significant amount of time. "*The method is inexpensive and fairly simple to run; however, it is difficult to achieve due to the large physical and cognitive effort required by the assessor*", said one of the evaluators. Therefore, the emotional evaluation performed used limited resources, but demanded time, especially of inexpert evaluators.
- The cognitive appraisals and behavioral components are used in the HCI area, especially in usability tests. However, they should also be considered as part of the emotions studies in HCI. The cognitive assessment is not usually discussed in emotional assessment techniques [24]. Nevertheless, it can be quite useful in the identification of design factors and user interface elements responsible for eliciting emotions. This is possible because the evaluation of cognitive activity changes quickly when new information is presented [21]. Thus, the designer can identify at interaction time, the reason/cause that led the user to perform a specific assessment (which can be expressed for example, through vocal expressions) of an event responsible for eliciting an emotion.
- Regarding behavioral tendencies, the fact that users had an average time required to complete the task, as well as the number of requests for help, greater than estimated, suggests that negative emotions elicited might have contributed to the obstruction of task execution.

Finally, the proposed approach supports the evaluation of the users' emotional responses due to the interaction with information systems. The evaluation led to a final assessment of the system. However, the methods could also be applied to assess each user's emotional state. Therefore, through flexible and more accessible design solutions, users with low literacy levels and less experience with technology could have better emotional experiences.

7 Conclusions

This paper presented a hybrid approach to the emotional evaluation of information systems. The approach is based on Scherer's model [1], and the evaluation methods were selected and adapted to measure the user's emotional response for the five components. A feasibility study was conducted considering a group of elderly users using tablet devices. The results suggest that the proposed approach can be easily applied, and moreover, it is relatively inexpensive.

The feasibility study also suggested that the approach is able to evaluate the users' emotional responses to the interaction, considering the software and the hardware used. In the elderly users' cases, less experience with tablets certainly influenced the final result. Moreover, the methods used also allowed the evaluation of a web information system. Future research can be performed to determine a better combination of methods for specific system domains. As Yusoff and Salim [12] noted for games, for instance, there are various physiological features that can be measured and related to the emotional response.

Future work will consider applying this hybrid approach to identify the emotional state of young people and adults during the interaction with information systems in order to analyze the degree of emotional experience obtained between and among groups of users. Another study to be done, deals with the identification of specific emotional responses considering specific contexts of interaction. The objective is to enable the designer to identify design elements and factors responsible for causing specific emotional responses. Therefore, identifying the users' emotional states, we intend to improve the design solutions to create interfaces that focus on satisfaction and emotional aspects of these users.

Acknowledgements. This work was funded by CAPES - Brazil. The authors thank colleagues from LIFeS-UFSCar for their insightful comments. The authors especially thank the elderly users, who gladly received the research group and participated in the feasibility study, the designers and the referees for their contributions.

References

1. Scherer, K.R.: On the Nature and Function of Emotion: A Component Process Approach. In: Scherer, K.R., Ekman, P. (eds.) Approaches to Emotion, pp. 293–317. Lawrence Erlbaum, Hillsdale (1984)
2. Cristescu, I.: Emotions in human-computer interaction: the role of nonverbal behavior in interactive systems. Revista Informática Económica **2**(46), 110–116 (2008)
3. Lim, Y.-K., Donaldson, J., Jung, H., Kunz, B., Royer, D., Ramalingam, S., Thirumaran, S., Stolterman, E.: Emotional experience and interaction design. In: Peter, C., Beale, R. (eds.) Affect and Emotion in Human-Computer Interaction,LNCS, vol. 4868, pp. 116–129. Springer, Heidelberg (2008)
4. Mahlke, S., Minge, M.: Consideration of multiple components of emotions in human-technology interaction. In: Peter, C., Beale, R. (eds.) Affect and Emotion in Human-Computer Interaction, LNCS, vol. 4868, pp. 51–62. Springer, Heidelberg (2008)

5. Scherer, K.R.: What are emotions? And how can they be measured? Soc. Sci. Inf. **44**(4), 695–729 (2005) (SAGE Publications)
6. Beale, R., Peter, C.: The role of affect and emotion in HCI. In: Peter, C., Beale, R. (eds.) Affect and Emotion in Human-Computer Interaction. LNCS, vol. 4868, pp. 1–11. Springer, Heidelberg (2008)
7. Norman, D.A.: Why We Love (or Hate) Everyday Things. Basic Books (2004)
8. Piccolo, L.S.G., Hayashi, E.C.S., Baranauskas, M.C.C.: The evaluation of affective quality in social software: preliminary thoughts. In: II WAIHCWS, IHC2010, pp. 29–38 (2010)
9. Hayashi, E., Neris, V., Baranauskas, C., Martins, M.C., Piccolo, L., Costa, R.: Avaliando a qualidade afetiva de sistemas computacionais interativos no cenário brasileiro. In: Usabilidade, Acessibilidade e Inteligibilidade Aplicadas em Interfaces para Analfabetos, Idosos e Pessoas com Deficiência: Resultados do Workshop. Campinas: CPqD, pp. 55–62 (2009)
10. Lera, E., Domingo, M.G.: Ten emotion heuristics: guidelines for assessing the user's affective dimension easily and cost-effectively. In: BCSHCI '07 Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI…but not as we know it, 2, pp. 163–166. British Computer Society, Swinton (2007)
11. Russell, J.A.: Affect grid: a single-item scale of pleasure and arousal. J. Pers. Soc. Psychol. **57**(3), 493–502 (1989)
12. Yusoff, N.M., Salim, S.S.: SCOUT and affective interaction design: evaluating physiological signals for usability in emotional processing. In: Second International Conference on Computer Engineering and Technology (ICCET), 2010, pp. 201–205 (2010)
13. Picard, R.W.: Emotion reserach by the people, for the people. Emot. Rev. **2**(3), 250–254 (2010)
14. Ortony, A., Clore, G.L., Collins, A.: The cognitive structure of emotions. Press Syndicate of University of Cambridge, Cambridge (1988)
15. Osgood, C.E., May, W.H., Miron, M.S.: Cross-Cultural Universals of Affective Meaning, 1st edn. University of Illinois Press, Urbana (1975)
16. Russell, J.A.: Pancultural aspects of the human conceptual organization of emotions. J. Pers. Soc. Psychol. **45**(6), 1281–1288 (1983)
17. Desmet, P.M.A.: A multilayered model of product emotions. Des. J. **6**(2), 4–11 (2003)
18. Russel, J.A.: A circumplex model of affect. J. Pers. Soc. Psychol. **8**(6), 1161–1178 (1980)
19. Shami, N.S., et. al.: Measuring affect in HCI: going beyond the individual. In: CHI EA '08 CHI '08 Extended Abstracts on Human Factors in Computing Systems (2008)
20. Longhi, M.T., Reategui, E.B., Bercht, M., Behar, P.A.: Um estudo sobre os Fenômenos Afetivos e Cognitivos em Interfaces para Softwares Educativos. In: Ciclo de Palestras Novas Tecnologias na Educação, vol. 9 (2007)
21. Scherer, K.R.: Emotion. In: Introduction to Social Psychology: A European Perspective, vol. 3, pp. 151–191. Blackwell, Oxford (2000)
22. Palen, L., Bödker, S.: Don't get emotional. In: Peter, C., Beale, R. (eds.) Affect and Emotion in Human-Computer Interaction. LNCS, vol. 4868, pp. 12–22. Springer, Heidelberg (2008)
23. Demir, E., Desmet, P.M.A., Hekkert, P.: Appraisal patterns of emotions in human-product interaction. Int. J. Des. **3**(2), 41–51 (2009)
24. Laurans, G., Desmet, P.M.A., Hekkert, P.P.M.: Assessing emotion in interaction: some problems and a new approach. In: International Conference on Designing Pleasurable products and Interfaces, pp. 13–16 (2009)
25. Desmet, P.M.A.: Measuring emotions: development and application of an instrument to measure emotional responses to products. In: Blythe, M.A., Monk, A.F., Overbeeke, K.,

Wright, P.C. (eds.) Funology: From Usability to Enjoyment, pp. 111–125. Kluwer Academic Publishers, Dordrecht (2003)
26. Geneva Appraisal Questionnaire Format, Development, and Utilization. http://www.affective-sciences.org/system/files/page/2636/GAQ_English.PDF
27. Someren, M.W., Barnard, Y.F., Sandberg, J.A.C.: The Think Aloud Method: A practical Guide to Modeling Cognitive Processes. Academic Press, London (1994)
28. Lefèvre, F., Lefèvre, A.M.C.: Discurso do sujeito coletivo - Um novo enfoque em pesquisa qualitativa (Desdobramentos). Livraria Resposta (2005)
29. Lang, P.J.: The cognitive psychophysiology of emotion: Fear and anxiety, pp. 131–170. Lawrence Erlbaum, Hillsdale (1985)
30. Reijneveld, K., Looze, M., Krause, F., Desmet, P.: Measuring the emotions elicited by office chairs. In: DPPI '03 Proceedings of the 2003 International Conference on Designing Pleasurable Products and Interface, pp. 6–10 (2003)
31. Izard, C.E.: The Psychology of Emotions. Plenum Press, New York (1991)
32. Mehrabian, A., Russell, J.A.: An Approach to Environmental Psychology. MIT, Cambridge (1974)
33. Russell, J.A., Mehrabian, A.: Evidence for a Three-Factor Theory of Emotions. J. Res. Pers. **11**, 273–294 (1977)
34. Watson, D., Clark, L.A., Tellegen, A.: Development and validation of brief measures of positive and negative affect: the PANAS scales. J. Pers. Soc. Psychol. **54**(6), 1063–1070 (1988)
35. Watson, D., Clark, A.: The PANAS-X: Manual for the Positive and Negative Affect Schedule - Expanded Form. Psychology Publications, New York (1999)
36. Cahour, B., Salembier, P., Brassac, C., Bouraoui, J.L., Pachoud, B., Vermersch, P., Zouinar, M.: Methodologies for evaluating the affective experience of a mediated interaction. In: Workshop on Innovative Approaches to Evaluating Affective Interfaces, CHI 2005, April 2–7, Portland, Oregon (2006)
37. Ekman, P., Friesen, W.V., Hager, J.C.: The Facial Action Coding Systems, 2nd edn, Salt Lake City: Research Nexus eBook. Weidenfeld & Nicolson, London (world) (2002)
38. Zaman, B., Smith, T.S.: The FaceReader: Measuring instant fun of use. In: NordiCHI 2006, pp. 14–18 (2006)
39. Axelrod, L., Hone, K.: Measuring affect: differentiating positive affect and politeness. In: CHI 2008 Workshop (2008)
40. Lichtenstein, A., Oehme, A., Kupschick, S., Jürgensohn, T.: Comparing two emotion models for deriving affective states from physiological data. In: Peter, C., Beale, R. (eds.) Affect and Emotion in Human-Computer Interaction. LNCS, vol. 4868, pp. 35–50. Springer, Heidelberg (2008)

Pragmatic Aspects of Collaborative Problem Solving: Towards a Framework for Conceptualizing Dynamic Knowledge

Rodrigo Bonacin[1(✉)], Heiko Hornung[2], Julio Cesar Dos Reis[3], Roberto Pereira[2], and M. Cecília C. Baranauskas[2]

[1]FACCAMP and CTI Renato Archer, Rodovia Dom Pedro I, Km 143.6, Campinas–SP 13069-901, Brazil
rodrigo.bonacin@cti.gov.br
[2]Institute of Computing, UNICAMP, Av. Albert Einstein no. 1251, Campinas–SP 13084-722, Brazil
heix@gmx.com, {rpereira, cecilia}@ic.unicamp.br
[3]CR SANTEC, Public Research Centre Henri Tudor, 2A rue Kalchesbrück 1852, Luxembourg, Luxembourg
julio.dosreis@tudor.lu

Abstract. Knowledge production in the Social Web can be understood as a dynamic socio-cultural process. Mechanisms that support users to explore this knowledge in an effective and efficient way may bring various benefits. However, the construction of these interaction mechanisms depends on further research on multidisciplinary fields. The interpretation of the content by users is influenced by meanings and intentions, as well as by the understanding of the evolution of these aspects over time. This paper analyses the evolution of meaning and intentions in collaborative problem solving. The analysis is informed by Semiotics and Speech Act theories. From this analysis, the paper proposes a conceptual framework for multidisciplinary research in three interconnected perspectives: interactive, conceptual and technical.

Keywords: Pragmatic web · Social network systems · Organizational semiotics · Knowledge evolution · Knowledge visualization

1 Introduction

The Social Web (SocWeb) [13] has a dynamic nature with respect to both content and enabled interactions. This dynamic nature affects the users' interpretations and intentions, and consequently their possibilities of interaction with the system. At the same time, the Semantic Web (SemWeb) [3] proposes to model the information in Web ontologies, aiming at enabling knowledge interpretation by artificial agents and by people. However, some of the main issues of the SemWeb include how it truly enables the connections of the Web of people who will use it, and how to turn "messy" human knowledge into a shared information space that is useful to everyone [14]. Therefore, an

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 410–426, 2013.
DOI: 10.1007/978-3-642-40654-6_25,

alignment of the SemWeb with SocWeb visions may bring benefits, but also open challenges in terms of requiring novel interaction methods and techniques.

In the context of the SocWeb, knowledge representation models applied to social software should take into account the dynamic aspects of the knowledge produced and exchanged by people in these systems. These models could enable richer user interactions by representing the evolution of the pragmatic aspects and their relations with the semantic ones. In this paper, we study the evolution of semantics and pragmatics as an integrated process.

In many cases, it is desirable to maintain the history of how content, interactions, as well as the meanings, intentions and interpretations evolve over time. In collaborative problem solving, for example, usually the interactions, as well as the rationale and history of the actions taken are as important as the solution itself. However, the interpretation of content generated during the collaborative process is dependent on the analysis of the evolution of the context, from the time when the content was created. Therefore, the associated knowledge representation models should evolve, maintain and present the evolutionary history in a proper way, associating meanings and intentions.

The construction of such systems relies on open research issues, requiring a multidisciplinary view to deal with. In this paper, research requirements are analyzed and situated in three interconnected perspectives: (1) *Interactive Perspective*: How users will actually visualize, make sense and interact with this dynamic content; (2) *Conceptual Perspective*: How to conceptually make sense and model the dynamic aspects of the knowledge (including meanings and intentions); and (3) *Computational Perspective*: How to implement and automate as much as possible the construction of the models.

In order to explore requirements for these views, the paper presents a study of dynamic aspects of pragmatics in messages exchanged during collaborative problem solving processes, within the special education domain. Two scenarios were explored: one in the "*Vila na Rede*" Social Network System (SNS) (www.vilanarede.org.br), which adopts a forum structure for questions and discussions, and the other within "*Yahoo! Answers*" (http://answers.yahoo.com/), which adopts the structure of multiple answers to a single question. The analysis was performed in three steps: the first step is related to a quantitative analysis of the interactions; the second step involves the examination of messages using the pragmatic communication analysis artifact [20]; and the third synthesizes the results and the exploratory analysis of interaction possibilities towards requirements for research in related fields. From the results of this study we extracted general requirements and challenges for the interactive, conceptual and computational perspectives. Based on the challenges raised, we present and discuss a preliminary framework for structuring future research.

The paper is organized as follows: Section 2 presents the background in Knowledge Visualization (KV), Knowledge Evolution (KE) and Pragmatic Web; Section 3 describes the study of pragmatics evolution in collaborative problem solving in SNS; Section 4 presents the challenges and discusses a conceptual framework to deal with the issues; Section 5 concludes the paper.

2 Background

In this section, we present background works in KV (Sect. 2.1), KE (Sect. 2.2) and Pragmatic Web (Sect. 2.3) as conceptual background for prospecting research requirements.

2.1 Knowledge Visualization

Visualization techniques are strategies to deal with the increasing quantity and complexity of subject matters in many domains [16]. According to Pampalk et al. [24] visualizations can make complex relationships easier to understand, and stimulate visual thinking.

Knowledge Visualization (KV) differs from Information Visualization (IV) techniques [9] in many aspects, including goals, benefits, content, or recipients [16]. In order to precisely define KV, many proposals discuss the differences of data, information, and knowledge, as well as try to reach a precise definition of the concept of knowledge. The concept of KV "in a strict sense is restricted to externalizing aspects of knowledge by the individual […] in a 'freestyle mapping mode'" [16, p. 7]. Also, the term KV has a focus on structured visualizations for the representation of conceptual knowledge [16].

In general, KV methods are required to make knowledge explicit and better usable, as well as to make sense of information structures [16]. Burkhard [5] proposes a KV Framework consisting of three perspectives (a Knowledge Type Perspective, a Recipient Type Perspective and a Visualization Type Perspective), which need to be considered when creating visualizations that aim to "transfer" knowledge. Different techniques have been proposed to address KV, such as: Scientific Charts, Concept Maps, Knowledge Maps, (Conceptual) Diagrams, Visual Metaphors, (Heuristic) Sketches [9, 16]. There are also ontology-based [18] as well as graph-based [22] approaches to KV. KV studies have also been applied to the SocWeb context [15].

In fact, an envisioned scenario for the Web [14] regarding the use of SemWeb artifacts within SocWeb environments poses new issues that have not yet been deeply explored in the KV literature. Therefore, investigations to clearly identify the challenges and new possibilities opened by this contemporary scenario are still required. For instance, there are dynamic aspects of the shared knowledge in social networks that are not addressed by the current KV methods. It is out of the scope of this paper to propose new KV techniques; however, we expect to contribute by clarifying interaction possibilities for KV in a dynamic SocWeb perspective.

2.2 Knowledge Evolution

The Ontology Evolution (OE) problem has mainly emerged with the use of ontologies in the context of the SemWeb. A well-accepted definition for OE is given by Stojanovic [30] as: "the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artifacts". Over the last years, distinct methods and approaches to organize the evolution steps have been proposed to treat the OE problem.

Stojanovic [30] proposes a six-step method that focuses on different aspects of the changes: (1) detecting, (2) representing, (3) defining its semantics, (4) implementing, (5) propagating and (6) validating. For each of these steps, various different approaches are proposed in the literature. Flouris et al. [10] present a survey on ontology change.

The OE problem has also been addressed and considered under different perspectives. For instance, approaches defined in ontology languages for the OE [2], ontology versioning [17], and community-based OE [19]. These approaches and methods for OE have resulted in the development of tools for supporting the OE process. Software applications such as *OntoStudio* (www.ontoprise.de/en/products/ontostudio) or *Protégé* (*protege.stanford.edu*) are generally augmented with additional functionalities through the use of plug-ins in order to support specific OE requirements. OE methods, techniques and tools are important for supporting the proposed conceptual and technical perspective.

In this work, we present requirements for the development of new KE techniques that consider the relations between semantics and users' intentions.

2.3 Pragmatic Web

According to Morris [23], pragmatics can be understood as the relationship between signs and humans. It concerns aspects such as intentions, communications, conversations and negotiations. While the areas of KV and KE focus on aspects related to the visualization of knowledge and formalisms that describe knowledge evolution, the Pragmatic Web is also concerned with the question of how knowledge is actually constructed and how it evolves during the collaboration among people that is mediated by Web artifacts. Originally proposed as an extension or a complement of the SemWeb, the Pragmatic Web addresses topics such as context and meaning negotiation in the Web [26, 28].

The Pragmatic Web perspective has been applied to a variety of research domains, e.g., multi-agent systems [25], interaction design [6], self-organizing communities of practice [8], or Web Services [21]. Pragmatic Web research is often rooted in different Information Systems research frameworks and theories, e.g., the Language/Action Perspective (LAP) [12, 31] or Organizational Semiotics (OS) [20].

The basic unit of analysis of LAP is a speech act. LAP subscribes to the notion that we perform actions through language. Thus, collaboration is coordinated by the performance of speech acts, which underlie socially determined rules [26]. In OS, basic units of analysis are affordances and agents. Initially introduced by Gibson [11], the concept of affordances was expanded by Stamper [29] to represent patterns of behavior that are governed by systems of norms in the physical and social world. Agents are entities (persons or groups of people) that can be attributed with responsibility. OS's basic ideas have been formulated as "there is no knowledge without a knower, and there is no knowing without action" [20, p. 26]. OS subscribes to the notion that knowledge about the world, and the underlying systems of norms are constantly changing.

Considering LAP and OS [7] as theoretical frames of reference, and having as object of study Web-mediated collaboration and meaning negotiation, the Pragmatic Web, thus, provides an important basis for this work.

3 Analyzing Dynamic Knowledge

The empirical data that has been analyzed in this work has been gathered during activities conducted in the context of a research project named TNR (www.nied.unicamp.br/tnr) in the domain of Web-mediated continuous learning of Brazilian special education teachers. Regarding the required training for special education teachers, the Ministry of Education defined an eighteen-month distance learning course for regular teachers. During this course, teachers learn to discuss a so-called "case" of students with special needs.

In Brazil, special needs are classified into 7 categories (visual, auditory, motor, intellectual impairment, intellectual giftedness, pervasive developmental disorder, multiple impairments), and in order to constitute a meaningful and representative group of 28 participants, 4 specialists of each category, from different parts of the country, who expressed a familiarity with the use of information and communication technology (ICT) were chosen randomly.

In the context of this project, the initial activities aimed at learning more about the way the 28 participants use Web systems, and how they engage in different forms of Web-mediated conversations. To this end, four consecutive activities were planned. Due to space limitations, the analysis presented in this work is based on two of the four activities, namely the discussions conducted in "*Vila na Rede*" and "*Yahoo! Answers*". *Vila na Rede* is an inclusive social network that permits to post "announcements" and to comment on them. Comments are displayed in a hierarchical structure below the announcement, and may contain text, pictures, audio or video. If the creator of the announcement authorizes it, other users may "collaborate", i.e., change or augment an announcement's text or media files. *Yahoo! Answers* is a Web system that permits a user to post a question and other users to post answers to that question or vote for the best answer.

Out of the 28 teachers, 16 participated in the two activities (9 participated in both activities, 6 only in *Yahoo! Answers* and 1 only in *Vila na Rede*). These teachers had no previous experience with the mentioned systems, but were already used to Web applications such as blogs, email, and forums. A "case" was posted in each system and teachers were asked to discuss and solve it.

3.1 Method

The objective of the present analysis is to investigate requirements and possibilities for a prospective computational mechanism that explores the dynamic aspects of content meanings and intentions of problem solving in SNSs. From the analysis of probable meanings and intentions of the written communication acts, we expect to explore new possibilities and extract requirements for research, design and implementation of such mechanisms, as well as to reveal research challenges to be overcome.

The first step of the three-step analysis regards the analysis of interaction including quantitative aspects. The interactions (e.g., comments, questions, answers) among users during the problem solving are enumerated and analyzed in a temporal order, resulting in an interaction graph. The activity of the network is also observed. One interaction may contain more than one message addressed to a receiver.

The second step involves the communication examination based on the pragmatic analysis presented in Liu [20]. We propose the use of this technique because it provides a structured way to analyze pragmatic aspects in messages. According to the Speech Act Theory (SAT) [1, 27] the acts are classified as *locutionary* (i.e., actual utterance and its ostensible meaning), *illocutionary* (i.e., propositional contents carrying intentions) and *perlocutionary* acts (i.e., effect on the addressee). As presented by Liu [20], a message can be divided into two parts: the content part of a communication act that manifests the meaning of the message as it is expressed in the proposition; and the function part of a communication act specifies the illocution that reflects the intention of the speaker. An interaction between users identified in the first step can be broken down into one or more messages. In this sense, for example, a long answer is considered one interaction unit, but it may contain more than one message unit from the communication act point of view.

In Liu [20] the illocutions are grouped into three dimensions: *time* (i.e., whether the effect is on the future or the present/past), *invention* (i.e., if the illocution used in a communication act is inventive or instructive, it is called prescriptive, otherwise descriptive), and *mode* (i.e., if it is related to expressing the personal modal state mood, such as feeling and judgment, then it is called affective, otherwise denotative). By using these dimensions, the illocutions are classified as: 1. *Proposal* (future, prescription and denotative), 2. *Inducement* (future, prescription and affective), 3. *Forecast* (future, description and denotative), 4. *Wish* (future, description and affective), 5. *Palinode* (present/past, prescription and denotative), 6. *Contrition* (present/past, prescription and affective), 7. *Assertion* (present/past, description and denotative), and 8. *Valuation* (present/past, description and affective).

Figure 1 shows the proposed evaluation form used in the second step. For each message, two analysts attributed continuous values from 0.0 to 1.0 for each dimension, for example, 0.0 for a message that the analyst judges to be totally denotative, and 1.0 for a totally affective one. An analyst might, for instance, attribute a confidence level smaller than 1 to a phrase (s)he could not classify confidently.

Based on the values attributed before, a predominant classification is attributed to each message, and optionally the analyst can also indicate a secondary (or alternative) classification. The analysts also identify aspects associated with the content part: the role of who performed the message ("speaker"), and the main object, affordance or proposition that the message refers to. This step was manually performed by the two analysts.

The third step is the synthesis of the results of step one and two. Graphs showing the evolution of each of the three dimensions of the illocutionary acts were produced

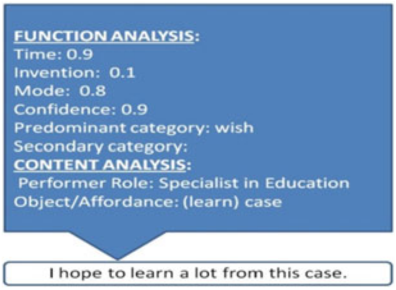


Fig. 1. An example of communication message analyzed using the proposed form.

to facilitate the analysis. A free exploratory analysis, in the format of a brainstorming, was performed aiming to investigate how the dynamic aspects could potentially contribute to interaction design for continuous learning, regarding mainly topics such as content recovery or search. Finally, a synthesis of the challenges and future research needs was performed.

3.2 Results

Figure 2 presents an interaction graph produced in the first step of the analysis of *Vila na Rede* case. The circles represent the 12 users (10 participating teachers and 2 facilitators from the research team) that performed at least one interaction, and the arcs represent their interactions. Circle sizes are proportional to the total number of interactions the respective users were involved in. The arc thickness is relative to the number of interactions performed by each pair of users.

By observing Fig. 2, it is possible to identify that the users V1 and V10 were the most active ones. In average, each of the 12 users performed 9.25 interactions, and approximately 77 % of all interactions where performed as an answer to a previous one. This can be interpreted as evidence that the problem solving and meanings were constructed in an interactive process, in which messages were constructed dynamically over the interpretation of previous messages.

In the second step, the content of the messages was analyzed using the results of step one as a starting point. Figure 3a, b, c present the evolution of the *time* (Fig. 3a), *invention* (Fig. 3b), and *mode* (Fig. 3c) dimensions during the problem solving

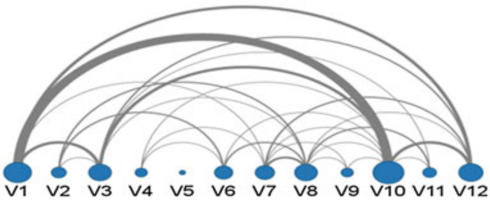


Fig. 2. Interaction Graph of *Vila na Rede* Case.

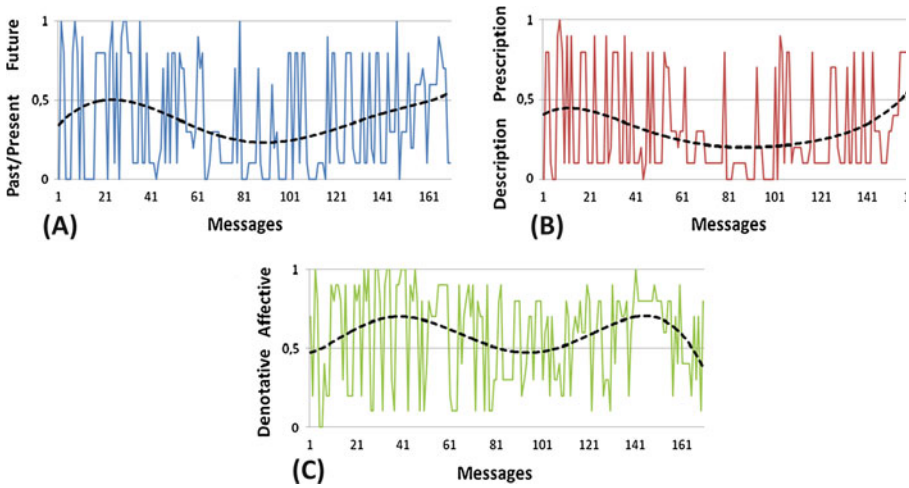


Fig. 3. Time, invention and mode dimension of the *Vila na Rede* case.

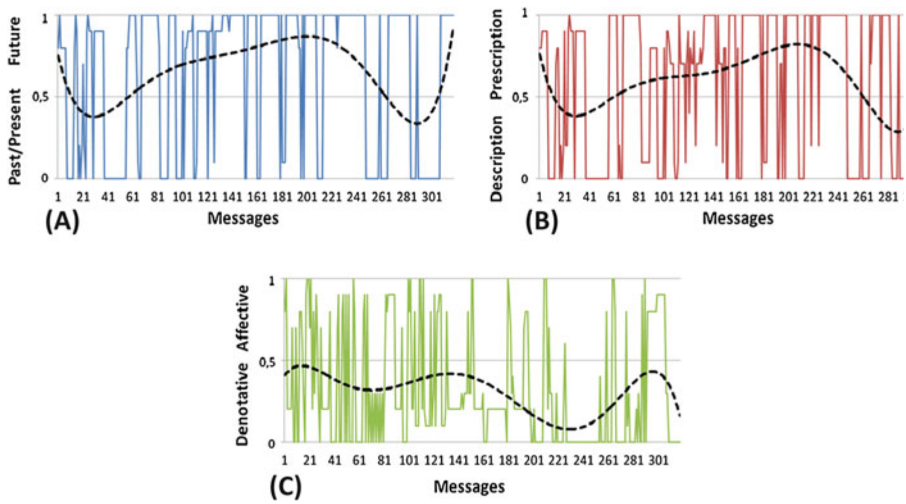


Fig. 4. Time, invention and mode dimension of the *Yahoo! Answers* case.

process in the *Vila na Rede* activity. Some interactions analyzed in the first step consisted of two or more illocutionary acts. A total of 170 illocutionary acts (contained in 110 interactions) were identified and plotted on the horizontal axis of Fig. 3a, b, c. The dashed lines in Fig. 3a, b, c represent the polynomial trend lines of each dimension. Looking at the trend line we can visualize the evolution of the dimensions in the problem solving process. For example, in Fig. 3b, in the interval 61 to 101, there is a predominance of descriptive messages, while after message 151, the prescriptive messages predominate.

Figure 4a, b, c present the evolution of the *time* (Fig. 4a), *invention* (Fig. 4b), and *mode* (Fig. 4c) dimensions during the problem solving process in the *Yahoo! Answers* activity. Each interaction analyzed in the first step consisted in average of 8 illocutionary acts. A total of 318 illocutionary acts (contained in 39 interactions) were identified and plotted on the horizontal axis of Fig. 4a, b, c. The vertical axis represents the respective dimension values attributed to each communication message using the form of Fig. 1. The dashed lines in Fig. 4a, b, c represent the polynomial trend lines of each dimension.

3.3 Prospecting Novel Interactive Mechanisms

When cross-referencing the function analysis with the content analysis, aspects regarding the problem solving processes can be observed. The content analysis links messages to the discussed issues (from the observed affordances). For instance, as marked in the object/affordance field of Fig. 1, the speaker was talking about the case under discussion. At each moment of the discussion, there is a single or at most few dominant topics. After determining these "dominant topics", they were cross-referenced with the graphs of the function analysis. Some major aspects observed during this third step of the analysis were:

– In *Vila na Rede*, at the beginning of the discussion, the specialists in education made assertions, some valuations and few requests about the case discussion itself. These aspects are especially interesting, for example, when users want to recover portions of dialogs that deal with discussion and planning.
– In *Vila na Rede*, the middle part of the discussion is dominated by valuations (present, description and affective) and assertions (present, description and denotative) regarding facts and alternatives for the case solution. This part of the dialog can potentially support the identification and visualization of facts and alternatives that support the solution.
– In *Vila na Rede*, at the end of the discussion, the specialists focused on a synthesis of a solution plan (future, prescription, denotative). This part of the solution is important to visualize the formulation of the solution itself.
– In *Yahoo Answers!*, at the beginning of the solution process, messages focused on requesting additional data about the case from the person who posted the question (i.e., proposals: future, prescriptive, denotative), this was followed by messages with assertions regarding the case (present, descriptive, denotative). This part of the solution processes is very important, for example, for someone who wants to understand the case, when some case details are not present in the primary question.
– In *Yahoo Answers!*, the middle part of the process focused on presenting solution plans (Special Education Pedagogic plans). Some plan proposals (future, descriptive, denotative) contained many items, and followed a structured format. This part of the processes is important for someone who wants to visualize the concurrent proposals, for example.

- In *Yahoo Answers!*, the end of the process focused on assertions (present, descriptive, denotative) and valuations (present, descriptive, affective) concerning the case and solutions. This was followed (at the very end) by the prescription (future, prescriptive, denotative) of a final plan. This part of the solution is important to visualize the final plan, as well as the valuations about the case and alternatives (e.g., some answers valuated the role of the family and the school in the described case, and proposed alternatives to work with this aspect).
- It was possible to visualize blocks of messages with the same (or close) values, e.g., messages 11-17 and 71-77 in Fig. 3a. Many blocks occurred simultaneously in more than one dimension and were related to the same affordance and/or were performed as answers to the same message. More data is required to understand the significance of these blocks for a problem solving process. However, these blocks are likely to represent correlated messages and once identified, they could be explored, for instance, by recovery and visualization techniques.
- By including the pragmatic analysis, a more refined classification of the messages is possible. For instance, it is possible to visualize and recover sequences of messages that valuate or judge one specific alternative.
- A more refined analysis of users' participation is possible. One possibility is to differentiate when a user requested information about one specific technique to solve a problem, or valuated alternatives, or did a lot of assertions about the techniques. This is especially important in social networks when we are looking for someone who has experience with a specific problem. Thus, it is possible to identify someone who commented about one topic, and also what the declared intention was (proposal, assertion, etc.).
- The identification of the illocutions can also be used as a parameter in the syntactic and semantic disambiguation process. For instance, the word "who" could be an interrogative pronoun in a request (e.g., who did that), but also a relative pronoun in an assertion (e.g., that's the guy who did this).
- Palinodes and contritions were the least frequent in the process of the analyzed cases. Nevertheless, they are extremely important in some situations. In the *Vila na Rede* case, one specialist proposed a synthesis of future actions, however in the subsequent messages she apologized for a mistake in one of the proposed actions.
- The analysis also revealed that there were assertions, valuations and proposals that were not related to the problem in focus, such as greetings and messages about the use of the computational systems. The identification of such messages might enable to filter, or even highlight the messages associated with the "core" problem (substantial messages). Another possibility is to filter "transversal" issues, such as the use of the functionalities of the systems (communication and control messages).
- Although comparisons between *Vila na Rede* and *Yahoo Answers!* are out of the scope of this paper, the pragmatic analysis also revealed important aspects regarding what forms of conversation or dialogue are supported in each system. For example, in *Vila na Rede*, questions and comments are displayed in a hierarchical structure. This resulted in less illocutions per interaction/post and more diverse illocutions types, while the discussion in *Yahoo Answers!* was predominated by

proposals to solve the posted question in long post messages that contained a high number of illocutions.

Starting from these possibilities we extracted requirements for the construction of features in SNSs that deal with and take advantage of the dynamic pragmatic aspects. A major requirement is the possibility for the users to visualize, explore and make sense of the dynamism of the pragmatic aspects of problem solving.

Users should be able to recover or filter parts of the solution processes using adequate parameters and be able to interact with the results in a useful way. For illustration purposes, consider the following hypothetic scenario: Someone wants to know what are the previous solutions for a specific problem discussed in the SNS, including, for example, the rationale and the evolution of the solution, who gave opinions and valuations, when, what were the alternatives and why they were not adopted (valuations about discarded alternatives), etc. How do users specify what they need; their information requirement? Should they directly specify the illocution classification, e.g., valuations about extra-scholar activities to support deaf students in the last 2 months? How to visualize the results? How to explore the results (e.g., select a term to show who was referred and when in the discussions)?

The questions presented are a small subset of possible issues that need to be explored. Most of these issues demand a multidisciplinary approach, which may involve techniques, methods and theories of various research fields.

4 A Framework for Conceptualizing Dynamic Knowledge

In this section, we present a preliminary research framework for conceptualizing pragmatic aspects of dynamic knowledge. This framework is the result of an exploratory investigation of areas, methods, techniques and technologies related to dynamic aspects of knowledge in the SocWeb. The scope of this section is not to provide a well-defined process or an exhaustive list of methods that can be directly translated into a solution. Rather, the framework presented here is intended to point out areas, technologies, and needs for deeper investigation.

Figure 5 presents the three interconnected perspectives of the proposed framework. Examples of candidate areas and technologies are shown for each perspective. The Figure shows that there is a significant overlap of the perspectives. Each item is classified according to the perspective where it is expected to contribute most. For example, in our proposed framework, SemWeb technologies are most relevant for the computational perspective, although they also might inform the conceptual or interactive perspective. Each perspective is discussed in the following subsections.

4.1 Interactive Perspective

In the interactive perspective, due to the diversity of the Web and potential users, communication becomes more and more complex and entails various design

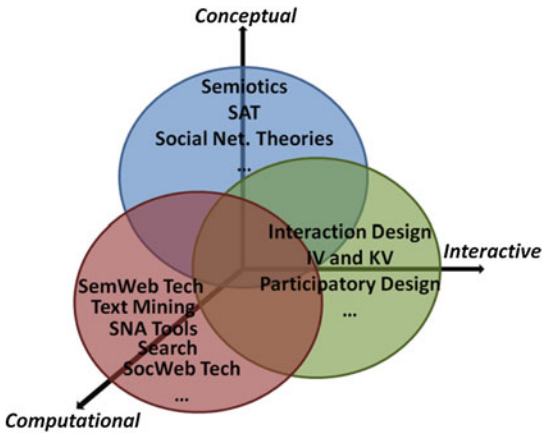


Fig. 5. Areas and Techniques in the three dimensions.

challenges. Two major challenges to be addressed are how to explore pragmatic aspects to: 1. support Web-mediated interaction in contexts where knowledge has to be constructed and evolves over time, and 2. provide interaction design solutions that enable the user to recover, explore and manipulate data and knowledge representation models. In this sense, new interaction design techniques have to be investigated.

During face-to-face communication, people use a variety of mechanisms, such as facial expressions, gestures, inflection, etc. These are determined by linguistic, social, cultural, and other aspects which delimit the participants' behavior and provide common ground. However, these mechanisms related to pragmatics are not always available (or visible) during written or computer-mediated communication. In problem solving, the interpretation of intentions might, among others, have an impact on the further solution process as well as on information retrieval and the reuse of solutions in future problems. Although there exist design solutions that enable users to explicitly express their intentions (e.g., by marking messages with images that express intentions), few works explicitly investigate the impact of pragmatic aspects in the Interaction Design (IxD) of Web applications for collaborative problem solving.

In this context, pragmatic aspects present in text messages exchanged among users have to be deeper investigated to determine appropriate IxD methods. It is particularly important to investigate how IxD designers can detect the users' needs and problems related to pragmatic aspects of communication, as well as to investigate how to generate design alternatives/candidates. The analysis of how the values of community members, designers and other stakeholders influence the IxD may indicate extra parameters in the proposition of user interface design methods.

IV and KV techniques are particularly important for providing means to visualize huge amounts of complex information (interactive perspective). The actual scenario in the Web, concerning SemWeb artifacts within SocWeb environments poses challenges that still need to be explored. It is necessary to articulate IV and KV with theories and novel frameworks for understanding knowledge (conceptual perspective).

The approaches so far explored in the literature are not able to accurately answer the question of whether the knowledge structures used really make sense for the target users. IV and KV techniques are still not able to deal with all the dynamic dimension of the knowledge. The domain is always changing over time, and it is necessary to cope with methods suited to better represent knowledge in a time-driven way. Therefore, theories and methodologies which can instrumentalize and inform novel ways to interact with the knowledge are still required.

Another issue to be addressed in the interactive perspective is how to capture in the system, the users' declared intentions. In some systems, users may declare their intentions through interactive interfaces, thus one of the main difficulties here is how to design appropriate interaction alternatives according to the context.

Participatory Design practices (and user-centered design) may facilitate the emerging of interaction alternatives. By working with users at design time, it is possible to determine how users make sense of the representations, and what the appropriate design options are, for example, regarding searching, filtering, visualizing and exploring information using pragmatic dimensions similar to the ones presented in Sect. 3.

4.2 Conceptual Perspective

In the conceptual perspective, one of the major questions is how to choose and employ adequate theories to support modeling the conceptual aspects of pragmatic knowledge. These models have to guarantee conceptual consistency and provide theoretical grounding to what is implemented in the computational models (computational perspective), and presented to users (interactive perspective). In principle, those theories are eligible to form the conceptual basis which might conceptualizes human actions and their effects, and which may be applied to computational and IxD-related contexts. There are three main challenges regarding the choice of theories and frameworks for the conceptual basis. First, the set of theories and frameworks has to be consistent regarding epistemological and ontological questions, to guarantee scientific rigor and methodological soundness. Second, the theories and frameworks need to be applicable to computational and IxD-related contexts. Third, the analyst must be able to apply the respective theories and frameworks, which might require a substantial initial amount of learning. In this work, we used Semiotics, LAP and SAT as conceptual bases, since they are suited to address our research questions and have been applied in computational and IxD related contexts.

Semiotics can provide us theories and a conceptual basis for future works (conceptual perspective). For example, the Norm Analysis Method (NAM) [20] offers a systematic way to understand and model the human behavior in society. However, some research issues have to be addressed, such as practical applications that provide better interactive and computational models, and the question of how to deal and model norms changes and evolution. Adequately providing support to conceptually model the norms evolution is crucial and poses a still open research challenge when dealing with the evolution of pragmatic aspects.

In the proposed framework, SAT and LAP are also part of the conceptual background regarding the analysis of pragmatics. Methods associated with SAT and LAP might be explored, and they may contribute to the development of methods and techniques towards a (semi-)automatic content analysis. The propositional attitudes (i.e., the effect of the illocutionary acts), which were out of the scope of this paper, should be investigated through the analysis of the perlocutionary acts. Theories developed in sociology and anthropology, e.g. those applied to Social Network Analysis (SNA) may complement the conceptual basis. Research on these theories, as well as their application in the design of systems, are needed to cope with dynamic aspects of the content produced and exchanged in the network. An interdisciplinary view is needed to provide models and techniques for representing and analyzing the social network behavior and its relation with the content exchanged in a dynamic basis.

4.3 Computational Perspective

In the computational perspective, two of the major issues to be addressed are: how to produce computer interpretable representations of the pragmatics, and how to automate the construction of these models to provide scalable solutions over time.

SemWeb technologies can be a starting point to the implementation of such solutions. For instance, concepts of Web Ontology Language (OWL) may have to be adapted or extended to deal with the pragmatic aspects. Moreover, in this context, the KE field has produced a set of methods and techniques to computationally deal with domain evolution through the use of SemWeb ontologies. However, research in this field is required to cope with the evolution of concepts and models related to pragmatics. Similarly to the evolution of semantics, e.g., the evolution of the meaning of concepts within a community, user's pragmatics also evolves. For example, at the beginning of a learning process, a person might ask questions to get answers. Later in the learning process, his/her focus might shift from asking questions to trying to respond simple questions and asking more complex questions. Still later, the person might start asking questions not for the purpose of getting answers, but for the purpose of instigating discussions and teaching other community members. Furthermore, intentions not only depend on the evolution of an individual, but also on other community members and their changes over time.

Text mining or information extraction techniques and tools are also particularly important for providing scalable solutions that include models or ontologies modeled from large volumes of content. It is not feasible to analyze thousands or millions of documents to produce ontologies. Text mining technologies and ontology learning techniques can utilize statistics, graphs, ranks and other representations that support the knowledge engineer in the construction of complex models, like ontologies, conceptual graphs and mind maps. Natural language processing and machine learning are usually highly required to support these techniques. However, compared to modeling exclusively based on semantics, pragmatic aspects introduce additional complexity that need to be faced. For example, a semantic model, including one that considers evolving semantics, might be able to determine possible meanings of a concept, including ambiguities, conflicting/contradicting meanings, etc. When considering pragmatics, the problem space becomes much more complex. A simple phrase, such as, “I hope to learn a lot from the case [discussion]” already can have multiple pragmatic meanings. For instance, it could be a simple assertion, an implicit promise to actively participate in the discussion, or – when taking into account non-sincere communication or rhetorical devices such as irony – a valuation in the sense of “this case is very simple, I certainly won’t learn anything new”. As in face-to-face communication, it depends on the listener or observer and other contextual factors to perceive these different meanings and to judge which one is most relevant.

Considering the challenges at hinted in this subsection, the existing computational models and techniques are far from being a reliable and precise regarding the representation of pragmatic aspects. Therefore a more realist solution should not be to explore new computational models that provide a high level of accuracy regarding the determination of pragmatic meanings, but to try to apply techniques to support designers, users and other stakeholders to adequately consider pragmatic aspects and their effects on interaction.

In this sense, a preliminary study was presented by [4]. It proposes an OWL ontology for representing pragmatic aspects in collaborative systems. The ontology is based on Liu’s approach for communication acts. It has three main classes: *Agent* (who performs an act), *Act* (the performed action), and *Behavior_Pattern* (the pattern that delineates the actions performed by one agent including the meaning interpretations). The communication acts are modeled by a set of classes and properties aiming to represent the three dimensions, i.e., time, invention and mode. This model enabled, for instance, a better understanding of the users in the problem discussion, and could be used for design or evaluation (e.g., for evaluating how different systems or different iterations of the same system support certain conversational features), or the use within communities of practice to review how a conversation had developed, what principal difficulties were encountered, etc. However, deeper studies are needed, including for example mechanisms for a semi-automatic classification of illocutions from the textual content.

5 Conclusions

Collaborative problem solving in SNS can potentially produce a huge amount of complex information. This information is of great value, for example to users in the context of continuous learning who want to recover information about past problems. However, the correct recovery and understanding of this information is influenced by pragmatic aspects that evolve over time. These aspects have a dynamic nature, and involve the examination of the intentions in the discussion, which is a complex issue.

In this paper, we presented a study of problem solving processes carried out in SNSs. The results pointed out new interaction possibilities, but also requirements for future research in related areas. The research needs were translated into a research framework structured in three perspectives: interactive, conceptual and computational. This framework indicates that novel approaches are necessary for dealing with the dynamic aspects of knowledge in these systems. Moreover, we stressed the importance of providing solutions that are able to conceptually and computationally articulate semantics and pragmatics as an integrated process over time. These solutions may demand, for example, studies in Organizational Semiotics and new KE and KV techniques. This framework also discusses IxD challenges concerning how to improve the user experience in problem solving systems by exploring a pragmatic view of the content exchanged by the users.

We hope that the proposed framework contributes with ideas towards facilitating meaningful Web-mediated interactions. Further work involves gathering more empirical data to test and refine the analysis method, coping with the challenges described in the previous sections, and investigating novel KV and KE methods based on the background disciplines.

Acknowledgements. This work is partially funded by CNPq (#560044/2010-0, #141058/2010-2), by Proesp/CAPES (#23038.01457/2009-11) and CAPES (#01-P-08503/2008). The authors also thank colleagues from InterHAD for insightful discussions.

References

1. Austin, J.L.: How to Do Things With Words. Oxford University Press, Oxford (1962)
2. Avery, J., Yearwood, J.: Dowl: A dynamic ontology language. In: Proceedings of the IADIS International Conference WWW/Internet 2003, Algarve, Portugal (2003)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. **284**(5), 34–43 (2001)
4. Bonacin, R., Dos Reis, J.C., Hornung, H., Baranauskas, M.C.C.: An ontological model for representing pragmatic aspects on collaborative problem solving. In: WETICE'12: Proceedings of the IEEE 21st International WETICE, Washington, DC, USA, pp. 444–449. IEEE Computer Society (2012)
5. Burkhard, R.: Learning from architects: The difference between knowledge visualization and information visualization. In: Eight International Conference on Information Visualization (IV04), London, July (2004)
6. Hornung, H., Baranauskas, M.C.C.: Towards a Conceptual Framework for Interaction Design for the Pragmatic Web. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part I, HCII 2011. LNCS, vol. 6761, pp. 72–81. Springer, Heidelberg (2011)
7. Cordeiro, J., Filipe, J.: Language action perspective, organizational semiotics and the theory of organized activity – a comparison. In: Proceedings of the Workshop DEMO. Tilburg, The Netherlands (2003)
8. de Moor, A., van den Heuvel, W.-J.: Web service selection in virtual communities. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences, Washington, DC, USA, 10 p. IEEE Computer Society (2004)
9. Eppler, M., Burkhard R.: Knowledge visualization. Towards a new discipline and its fields of application. Working Paper of Net Academy on Knowledge Media, St. Gallen (2004)
10. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: Classification and survey. Knowl. Eng. Rev. **23**(2), 117–152 (2007)
11. Gibson, J.J.: The Ecological Approach to Visual Perception. Houghton Miffin Company, Boston (1968)
12. Goldkuhl, G., Lyytinen, K.: A language action view of information systems. In: Proceedings of the 3rd International Conference on Information Systems, Ann Arbor, MI, pp. 13–29 (1982)

13. Gruber, T.: Collective knowledge systems: Where the social web meets the semantic web. J. Web Seman. **6**(1), 4–13 (2008)
14. Hendler, J., Berners-Lee, T.: From the semantic web to social machines: A research challenge for AI on the World Wide Web. Artif. Intell. **174**, 156–161 (2010). Elsevier
15. Hoetzlein, R.: The Organization of Human Knowledge: Systems for Interdisciplinary Research. Master's Thesis, University of California Santa Barbara, USA (2007)
16. Keller, T., Tergan, S.-O.: Visualizing Knowledge and Information: An Introduction. In: Tergan, S.-O., Keller, T. (eds.) Knowledge and Information Visualization. LNCS, vol. 3426, pp. 1–23. Springer, Heidelberg (2005)
17. Klein, M., Fensel, D.: Ontology versioning for the semantic web. In: Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford University, California, USA (2001)
18. Kuß, A., Prohaska, S., Rybak, J.: Using ontologies for the visualization of hierarchical neuroanatomical structures. In: 2nd INCF Congress of Neuroinformatics (2009)
19. De Leenheer, P., Meersman, R.: Towards Community-Based Evolution of Knowledge-Intensive Systems. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 989–1006. Springer, Heidelberg (2007)
20. Liu, K.: Semiotics in Information Systems Engineering. Cambridge University Press, New York (2000)
21. Liu, K.: Pragmatic Computing – A Semiotic Perspective to Web Services. In: Filipe, J., Obaidat, M.S. (eds.) ICETE 2007, Part I. CCIS, vol. 23, pp. 3–15. Springer, Heidelberg (2008)
22. Maseri, W., Mohamad, W., Embong, A., Mohd Zain, J.: Improve knowledge visualization through an interactive graph-based dashboard system with key performance indicator: A case study of university dashboard for higher education. In: National Conference on Software Engineering & Computer Systems 2007 (NaCES 2007), Legend Resort Kuantan (2007)
23. Morris, C.W.: Foundations of the Theory of Signs. International Encyclopedia of Unified Science, vol. 1, p. 2. University of Chicago Press, Chicago (1938)
24. Pampalk, E., Goebl, W., Widmer, G.: Visualizing Changes in the Structure of Data for Exploratory Feature Selection. In: SIGKDD'03, Washington, DC, USA (2003)
25. Paschke, A., Boley, H., Kozlenkov, A., Craig, B.: Rule Responder: RuleML-Based Agents for Distributed Collaboration on the Pragmatic Web. In: ICPW'07: Proceedings of the 2nd International Conference on Pragmatic Web, pp. 17–28. ACM, New York (2007)
26. Schoop, M., de Moor, A., Dietz, J.L.G.: The pragmatic web: A manifesto. Commun. ACM **49**(5), 75–76 (2006)
27. Searle, J.R.: A classification of illocutionary acts. Lang. Soc. **5**(1), 1–23 (1976)
28. Singh, M.P.: The pragmatic web. IEEE Internet Comput. **6**(3), 4–5 (2002)
29. Stamper, R.: Signs, information and systems. In: Holmqnist, B., et al. (eds.) Signs of Work Semiotics Information Processing in Organisations, Walter de Gruyter, New York (1996)
30. Stojanovic, L.: Methods and tools for ontology evolution. PhD. Thesis. University of Karlsruhe, Universitat Karlsruhe (TH), Institut AIFB, D-76128 Karlsruhe (2004)
31. Winograd, T., Flores, F.: Understanding Computers and Cognition: A New Foundation for Design. Addison-Wesley Longman Publishing Co., Inc., Boston (1987)

Part VII

Enterprise Architecture

Formalizing Meta Models with FDMM: The ADOxx Case

Hans-Georg Fill[1(✉)], Timothy Redmond[2], and Dimitris Karagiannis[1]

[1] Research Group Knowledge Engineering, University of Vienna, Vienna, Austria
{hgf, dk}@dke.univie.ac.at
http://www.dke.univie.ac.at

[2] Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, CA, USA
tredmond@stanford.edu
http://bmir.stanford.edu

Abstract. This paper contains an extended and improved version of the FDMM formalism presented at ICEIS'2012. FDMM is a formalism to describe how meta models and models are defined in the ADOxx approach as used in the Open Models Initiative. It is based on set theory and first order logic statements. In this way, an exact description of ADOxx meta models and corresponding models can be provided. In the paper at hand we extend the description of the formalism by illustrating how the mathematical statements can be used to support the implementation on the ADOxx platform. For this purpose we show how the FDMM constructs are mapped to statements in the ADOxx Library Language (ALL). As an example of the approach, the formalism and the mapping to ALL are applied to a modeling language from the area of risk management.

Keywords: Conceptual modeling · Meta modeling · Domain-specific modeling

1 Introduction

Conceptual models are today used in many areas of enterprise information systems [1–3]. Examples range from fields such as strategic management, business process and workflow management to enterprise architecture and software engineering. For these purposes a large variety of modeling languages have been developed and successfully applied in academia and industry [4]. When it comes to the sharing of such modeling languages and their corresponding models - as it has been recently promoted by the Open Models Intiative [5,6] - the exact description of a modeling language and the models is one of the most important tasks. These descriptions not only reflect the design choices made during the implementation of the language. They also permit to compare and learn from different implementations of a modeling language and support the interpretation of the models [7].

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 429–451, 2013.
DOI: 10.1007/978-3-642-40654-6_26,

In order to describe the building blocks of modeling languages it can be reverted to several types of *meta modeling* approaches [8]. These approaches provide the constructs necessary for describing the abstract and concrete syntax of a modeling language [9,10]. In this way they also define constraints and correctness criteria for creating valid models based on the definition of a modeling language. In the context of the Open Models Initiative, several projects[1] have reverted to the freely available and industry proven ADOxx[2] meta modeling approach. At its core, the ADOxx approach allows to specify the syntax of a modeling language together with its graphical representation. From these specifications, visual model editors are then created automatically [11].

For the description of ADOxx based modeling languages, it has so far either been reverted to natural language descriptions, e.g. [12,13], concrete implementations in the form of source code, e.g. [14,15] or visual representations, e.g. [16,17]. A formal description of the ADOxx meta modeling approach is so far not available. This is however necessary to analyze and evaluate how the syntax of a certain modeling language has been realized, compare ADOxx meta models and models to other meta modeling approaches such as GME, Ecore or ARIS cf. [8], derive suggestions for its enhancement and finally describe semantics for its further processing [9,18].

Therefore we propose the FDMM[3] formalism that is capable of describing the core constitutents of the ADOxx approach. FDMM aims to provide an easy to use formalism that does not require specialized mathematical knowledge and that is capable of expressing the implementation of ADOxx meta models and models. The paper is structured as follows: In Sect. 2 we will briefly discuss the foundations of modeling languages, meta models and models, the characteristics of the ADOxx approach and describe a running example for a modeling language from the area of enterprise information systems. Section 3 will describe the formalism including the necessary constraints and correctness criteria. In Sect. 4 the formalism will be applied to the sample modeling language. In Sect. 5 it will be shown how the formalism can be used as a basis for the implementation of the meta models in ADOxx using the ADOxx Library Language (ALL). The results of this implementation are then discussed in Sect. 6. Work related to the approach will be part of Sect. 7. The paper is concluded by an outlook on the future work in Sect. 8.

2 Foundations

In this section we will briefly define the terms *modeling language*, *meta model* and *model* and describe the main characteristics of the ADOxx meta modeling approach. Finally, we will present a running example by using a modeling language from the area of risk management.

[1] See http://www.openmodels.at/web/omi/omp

[2] ADOxx is a registered trademark of BOC AG.

[3] The acronym FDMM stems from: A Formalism for Describing ADOxx Meta Models and Models.

2.1 Modeling Language, Meta Model and Model

According to a framework proposed by [10], a modeling language is composed of a *syntax*, *semantics* and *notation*. The syntax specifies the elements and attributes of the language and the semantics assigns meaning to these constructs. In contrast to other frameworks, the notation is treated separately and is used to specify the visual representation of the language. The syntax further consists of an abstract syntax, which is represented by the *meta model*, and the concrete syntax, which is represented by a *model* [9,19,20]. The meta model can itself be described by a modeling language, i.e. the *meta modeling language*. Accordingly, the abstract syntax of the meta modeling language is represented by a *meta meta model* and the concrete syntax is represented by a meta model [8].

An example for these relationships is shown in Fig. 1: in the meta meta model the two entities *element* and *attribute* are defined. Additionally, a relation between the two entities is shown. On the meta model level the E_1 entity is defined as an element and the A_1 and A_2 entities as attributes that can be related to E_1. Finally, on the model level the entities ϵ_1 and ϵ_2 are defined as E_1 elements, the α_1 and α_2 entities as A_1 attributes and the β_1 and β_2 entities as A_2 attributes. The meta meta model thus defines which entities are provided for the specification of the abstract syntax of a modeling language in the form of a meta model. If the specification of the meta meta model is generic enough it can be used to specify a multitude of different modeling languages. A typical use case is then to automatically create model editors based on the static meta meta model specification and the dynamically adaptable meta model specifications.

In addition to *association mechanisms* for defining linkages between entities, meta meta models typically also provide *inheritance* and *containment* mechanisms [20]. By inheritance mechanisms, generalization and specialization relationships between entities of a meta model can be expressed to provide means for effecting polymorphic behaviors at model execution or interpretation time. This is required in particular for the design of algorithms that shall work on multiple, similar modeling languages without the need of particular adaptations: by defining an algorithm on a set of general entities that are shared by different meta models, the algorithm can be later bound automatically to entities that

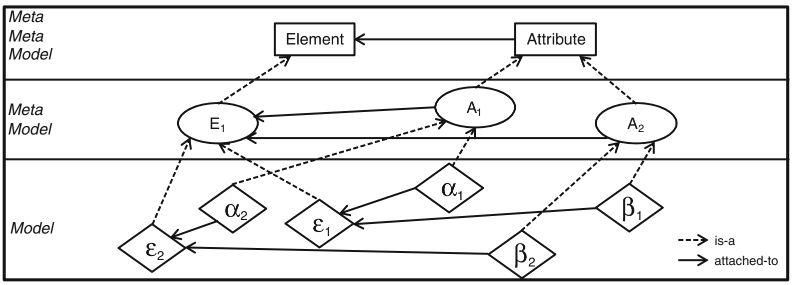


Fig. 1. Example for a Meta Meta Model, a Meta Model and a Model.

are inherited from these general entities. Containment mechanisms refer to the inclusion of a set of entities into another entity on the model level. This is typically used to specify model/diagram types or aggregations/nestings that group sets of entities.

2.2 The ADOxx Meta Modeling Approach

The ADOxx meta modeling approach has originally been conceived in the course of the development of the ADONIS business process management toolkit in 1995. Since then it has been successfully used in a large number of academic and commercial projects by more than 1000 customers worldwide [21,22]. The basic building blocks of its meta meta model are *classes* and *relationclasses* that are complemented with *attributes* [22]. Classes are organized in the form of an *inheritance hierarchy* so that the attributes of a super-class are inherited by its sub-classes in the sense of standard object orientation principles. Relationclasses are defined by their *from-class* and *to-class* attributes to specify valid instances of source and target classes. These relations can be complemented with cardinality constraints to limit the number of participating instances.

For collections of classes and relationclasses a containment mechanism in the form of *model types* is available. Model types define the context for the instantiation of classes and relationclasses in the form of *models*. Therefore, when creating a model, at first a model type has to be selected to specify which classes and relationclasses are valid in a model. Subsequently, these classes and relationclasses are instantiated as part of the model.

Besides the standard data types such as *integer*, *string*, and *double*, *enumeration* attributes are available in ADOxx that contain *pre-defined* values that can only be selected but not modified by a user during modeling. Furthermore, attributes can also be of two special types: attributes of the type *expression* contain strings in a proprietary expression grammar for automatically calculating the value of an attribute. Attributes of the type *interref* allow linking the instance of a class to another class instance of the same or of a different model instance or linking it to the same or a different model instance itself. The graphical representation of the instances of classes and relationclasses is specified via the particular string attribute named *GRAPHREP*. This attribute permits to specify context-dependent graphical representations for the classes and relationclasses, again based on a proprietary grammar - cf. [23]. Although an inherent part of the ADOxx approach, the graphical representation can thus be modified independently from the other entities.

With these characteristics, the following requirements were defined for a formalism that can describe the concepts of the ADOxx meta modeling approach: It should permit a formal description of the approach that is easy to handle and thus suitable for a wide range of users. Therefore, the formalism should focus on the core constituents to enable the description of arbitrary modeling languages that have been implemented using the ADOxx approach. It should however be extensible to allow its further development and future refinement.

2.3 Running Example: The 4R Modeling Language

As a running example we will revert to an existing modeling language in enterprise information systems from the area of governance, risk, and compliance (GRC). This example will first illustrate how meta models and models are typically used in information systems to create domain conceptualizations. In Sect. 4 we will revert to it again for showing the application of the FDMM formalism.

In the last years particular consideration has been given to the management of risks and regulatory compliance together with their effects on returns and corresponding reporting requirements of enterprises. In line with these developments, the framework for *integrated enterprise balancing* has been derived [13,24,25]. The goal of this framework is to govern business activities with organization-wide consistent return and risk indicators. As a foundation, it is necessary to provide a common data basis that represents information from the areas of *r*isk, *r*eturn, *r*egulation, and *r*eporting - the so-called 4R information architecture. For acquiring this information, the *4R modeling language* [13] and the *4R situational method* for implementing such solutions in an organization were developed [25]. In its original form, the central parts of the 4R modeling language were illustrated by an extension of a graph based formalism. The corresponding realization using the ADOxx meta modeling approach was however described in natural language.

The meta model of the 4R modeling language, as it was specified in ADOxx, contains the following model types [13]: the *4R portfolio model*, the *4R business process model*, and the *4R organizational model*. Briefly summarized, the portfolio model type is used to describe multi-dimensional aggregations of the risks, returns and correlations of business transactions in regard to their relations to products and customers [24]. The single business transactions in this model can be linked to instances of the 4R business process model type. This second model type extends the process modeling language of business graphs [26] with elements, relations, and attributes for representing events, aggregations of events and their influence on the properties of process activties. The meta model is complemented with a 4R organizational model for representing actors, organizational units, resources, and roles that fulfill tasks in a business process.

For the implementation of the 4R model types on ADOxx the following classes and relationclasses together with several attributes were specified to represent the risk and return figures of business transactions and the underlying risk-affected business processes:

- for the portfolio model type the class *business transaction* and the relationclass *relates business transaction*,
- for the 4R business process model the super class *FlowObject* and as sub classes of this class: *Start*, *Decision*, *SubProcess*, *Activity*, *Parallelity*, *Join*, and *End*. Additionally the classes: *4R risk aggregation*, and *4R event*; the relationclasses: *subsequent*, *4R aggregation relation*, and *4R influences relation*
- for the 4R organizational model the classes: *actor*, *organizational unit*, *resource*, and *role*; and the relationclasses: *uses resouce*, *belongs to unit*, *has role*, and *has resource*

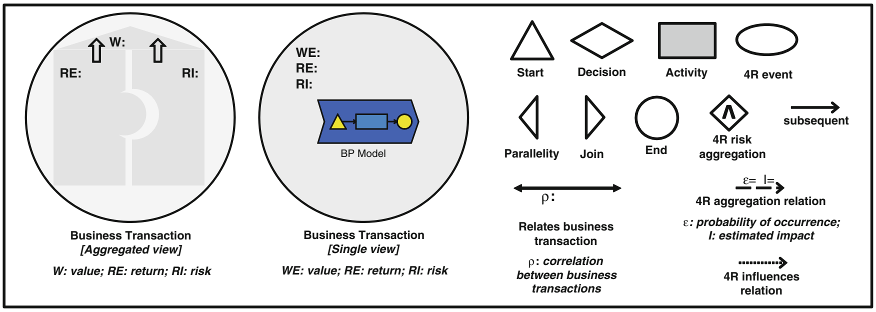


Fig. 2. Symbols of the 4R Portfolio Model Type and the 4R Business Process Model Type.

To apply the FDMM formalism we will later regard the 4R portfolio model and the 4R business process model type in more detail. The classes and relationclasses of these model types are represented by the sets of symbols as shown in Fig. 2.

3 The FDMM Formalism

As stated in the introduction, the goal of this paper is to develop a formalism that is capable of describing the core constituents of ADOxx meta models and models as well as the criteria for valid models based on the meta model definitions. FDMM is therefore not a formalization of all aspects of the ADOxx approach, but a formalism that aims to support users of ADOxx to describe their meta models and models in a formal way. For the development of the formalism we reverted both to literature sources on the ADOxx meta modeling approach as well as existing implementations [2,11,23]. During the development we aimed for keeping the formalism as simple as possible while not sacrificing any of the core concepts of the approach.

3.1 Definition of Meta Models

At first we define the basic constituents of meta models **MM** which can then be used to derive model instances **mt**. We define a meta-model to be a tuple of the form

$$\mathbf{MM} = \langle \mathbf{MT}, \preceq, \mathrm{domain}, \mathrm{range}, \mathrm{card} \rangle \tag{1}$$

where

– **MT** is the set of model types. We have

$$\mathbf{MT} = \{\mathbf{MT}_1, \mathbf{MT}_2, \ldots, \mathbf{MT}_m\} \tag{2}$$

where $\mathbf{MT}_i$ in turn is a tuple,

$$\mathbf{MT}_i = \langle \mathbf{O}_i^T, \mathbf{D}_i^T, \mathbf{A}_i \rangle \tag{3}$$

consisting of the set of object types $\mathbf{O}_i^T$, a set of data types $\mathbf{D}_i^T$, and a set of attributes $\mathbf{A}_i$. In this way ADOxx classes and relationclasses are uniformly represented as object types. As will be described below we will use the attributes $\mathbf{A}_i$ also for expressing associations between object types. When we describe the instantiation of the meta model in Sect. 3.2 in model instances, the attributes will map the instantiation of object types to the instantiation of either object types or data types. This permits us to represent the ADOxx concepts of relationclasses and interref relations in the same way. However, this also means that, if for example directed relations are required, the notions of source and target object types have to be added when specifying a particular meta model. The object types, data types, and attributes are part of their respective total sets:

$$\mathbf{O}^T = \bigcup_j \mathbf{O}_j^T, \mathbf{D}^T = \bigcup \mathbf{D}_i^T, \mathbf{A} = \bigcup \mathbf{A}_i \tag{4}$$

- $\preceq$ is an ordering on the set of object types, $\mathbf{O}^T$. If $o_1^t \preceq o_2^t$ we say that the object type o_1^t is a subtype of the object type o_2^t. Thereby the inheritance hierarchy of ADOxx classes can be expressed. Due to the generic definition, this ordering can also be used for relationclasses: as ADOxx requires a from-class and a to-class attribute for relationclasses, a generic object type with these attributes can be defined and used for the definition of subtypes that inherit these attributes.
- the domain function maps attributes to the power set of all object types, i.e.

$$\text{domain} : \mathbf{A} \rightarrow \mathcal{P}(\bigcup_j \mathbf{O}_j^T) \tag{5}$$

  The domain function will constrain what objects an attribute can map in the model instances. It is therefore used to attach attributes to a particular set of object types. In regard to ADOxx this corresponds to the assignment of ADOxx attributes to classes and relationclasses and the definition of an endpoint of an ADOxx relationclass.
- The range function maps an attribute to the power set of all pairs of object types and model types, all data types, and all model types

$$\text{range} : \ \mathbf{A} \rightarrow \mathcal{P}\left(\bigcup_j (\mathbf{O}_j^T \times \{\mathbf{MT}_j\}) \cup \mathbf{D}^T \ \cup \mathbf{MT}\right) \tag{6}$$

  In the model instances, the range function will constrain what values an attribute can take. For the definition of a meta model it is thus used to specify the type of an attribute. I.e. whether the attribute has the function of specifying the relation of an instance of an object type in a model instance to either: *a.* the instances of object types in a particular model type, *b.* instances of data types or *c.* instances of model types. Case *a.* thus corresponds to both the relationclass and interref concepts in ADOxx that have an instance of a class as a target, case *b.* to the ADOxx attribute concepts except interref attributes, and case *c.* to the ADOxx interref attribute concept that has an instance of a model type as a target.

– The card function maps pairs of object types and attributes to pairs of integers

$$\text{card} : O^T \times A \rightarrow \mathcal{P}(\mathbb{N} \times (\mathbb{N} \cup \{\infty\})) \tag{7}$$

where $\mathbb{N}$ is the set of non-negative integers. In the model instances the card function will constrain how many attribute values a object can have. In regard to the different types of attributes this thus permits to specify how many instances of object types, of data types or of model types an attribute can contain. When comparing this to the ADOxx approach, the card function determines whether the value of an attribute corresponds to either: *a.* a target of a relationclass - that can only be one distinct class, *b.* the target of an interref attribute that can have multiple values or *c.* the instance of a datatype that can also have multiple values, which corresponds to the enumeration attribute type in ADOxx.

In addition we define the following correctness criteria for meta models: The sets of object types, data types, and attributes have to be pairwise disjoint

$$\mathbf{O}^T \cap \mathbf{D}^T = \emptyset,\ \mathbf{O}^T \cap \mathbf{A} = \emptyset,\ \mathbf{D}^T \cap \mathbf{A} = \emptyset \tag{8}$$

This follows from the fact that in mathematical terms we have so far only been defining various sets that could overlap. In addition, for any attribute a that is part of the attribute set $\mathbf{A}_i$ of the i-th model type - see Eq. 3, the domain function for that attribute must point to any of the object types in that model type

$$a \in \mathbf{A}_i \Rightarrow \text{domain}(a) \subseteq \mathbf{O}_i^T \tag{9}$$

That ensures that attributes that are related by the domain function to a certain object type are part of the same model type definition. This corresponds directly to the ADOxx approach in the way that all concepts that are relevant for the definition of models are grouped within the context of a model type.

3.2 Instantiation of Meta Models

We will now describe the instantiation of a meta model. The instantiation of a meta model essentially describes the mapping of the model types, object types, and data types to model instances, objects, and data values together with a set of triples. Thus, an instantiation of a metamodel **MM** will be a tuple

$$\left\langle \mu_{\mathbf{mt}}, \mu_{\mathbf{O}}, \mu_{\mathbf{D}}, \mathcal{T}, \beta \right\rangle \tag{10}$$

where

– $\mu_{\mathbf{mt}}$ is a one-to one mapping from model types to the power set of model instances:

$$\mu_{\mathbf{mt}} : \mathbf{MT} \rightarrow \mathcal{P}(\mathbf{mt}) \tag{11}$$

Thereby it is defined that a model instance must be of one specific model type and that there may be several instances of one model type.

- $\mu_{\mathbf{O}}$ is a function taking the object types in a given model type to collections of objects:

$$\mu_{\mathbf{O}} : \bigcup_j \left(\mathbf{O}_j^T \times \{\mathbf{MT}_j\}\right) \rightarrow \mathcal{P}(\mathbf{O}) \tag{12}$$

where

$$\mathbf{O} = \bigcup_j \mu_{\mathbf{O}}(\mathbf{O}_j^T \times \{\mathbf{MT}_j\}) \tag{13}$$

Thereby, the objects are defined as instances of an object type $\mathbf{O}_j^T$ that is part of a particular model type $\mathbf{MT}_j$ - see also Eq. 3. The addition of the model type is necessary as object types may be part of multiple model types and in the ADOxx approach objects can only occur within a model instance. Sometimes it is convenient to create an object type which is meant to be subtyped but which is not meant to be directly instantiated. The purpose of such a type is to capture information that is common to all the subtypes. Such a type is called an *abstract type* and we can define what it means to be an abstract type based on the definitions above. An object type $o^t \in \mathbf{O}^T$ is said to be abstract if for all model types $\mathbf{MT}_i$ which contain the object type o^t $(o^t \in O_i^T)$ we have

$$\mu_{\mathbf{O}}(o^t, \mathbf{MT}_i) = \bigcup_{o_1^t \neq o^t, o_1^t \preceq o^t} \mu_{\mathbf{O}}(o_1^t, \mathbf{MT}_i). \tag{14}$$

That is to say that all the objects that instantiate o^t must instantiate o^t through one of its subtypes. In terms of ADOxx the notion of abstract types corresponds to super classes of which one or more of their sub classes are included in a model type but who cannot be instantiated themselves.

- $\mu_{\mathbf{D}}$ maps data types to a power set of data objects

$$\mu_{\mathbf{D}} : \mathbf{D}^T \rightarrow \mathcal{P}(\mathbf{D}) \tag{15}$$

The data types are not further constrained. It is thus left to the user of the formalism to ensure the correct content of a type, e.g. whether an 'integer' type contains only integer numbers. The formalism will only ensure that a data object is assigned a type that is valid in a particular context.

- $\mathcal{T} \subseteq \mathbf{O} \times \mathbf{A} \times (\mathbf{D} \cup \mathbf{O} \cup \mathbf{mt})$ is a set of triples. These triples will later be used to describe the contents of model instances.
- $\beta : \mathbf{mt} \rightarrow \mathcal{P}(\mathcal{T})$. This map describes how the triples are assigned to the model instances.

We will additionally define a collection of correctness constraints on the instantiation of the meta model. These constraints fall into two categories: disjointness constraints that describe how a model instance is partitioned and domain/range/cardinality constraints that constrain how attributes can map objects to other objects and data values.

The following constraints define the disjointness and partitioning constraints that must be enforced for the various parts of the meta model instantiation:

- The instances of object types and the instances of datatypes must be disjoint, i.e. instances of object types and instances of datatypes cannot be the same.

$$\mu_{\mathbf{O}}(o^t, MT_j) \cap \mu_{\mathbf{D}}(d^t) = \emptyset \tag{16}$$

- The instances of two object types are disjoint if either the object types are disjoint or if their model types to which they belong are disjoint, i.e. the formalism does not permit the instantiation from multiple object types nor a 'reuse' of objects of the same object type for different model instances:

$$i \neq j \vee o_1^t \neq o_2^t \Rightarrow \mu_{\mathbf{O}}(o_1^t, \mathbf{MT}_i) \cap \mu_{\mathbf{O}}(o_2^t, \mathbf{MT}_j) = \emptyset \tag{17}$$

- For two different model types $\mathbf{MT}_i$ and $\mathbf{MT}_j$ also the corresponding model instances must be disjoint, i.e. also for model instances no instantiations from multiple model types are allowed:

$$\mathbf{MT}_i \neq \mathbf{MT}_j \Rightarrow \mu_{\mathbf{mt}}(\mathbf{MT}_i) \cap \mu_{\mathbf{mt}}(\mathbf{MT}_j) = \emptyset \tag{18}$$

- Every element of the set of model instances **mt** has to be derived from a model type, i.e. there cannot be model instances without a corresponding model type:

$$\mathbf{mt} = \bigcup \mu_{\mathbf{mt}}(\mathbf{MT}_j) \tag{19}$$

- For two different data types it must follow that also their instances are disjoint, i.e. also for data types it is not allowed that instances can be derived from multiple types:

$$d_1^t \neq d_2^t \Rightarrow \mu_{\mathbf{D}}(d_1^t) \cap \mu_{\mathbf{D}}(d_2^t) = \emptyset \tag{20}$$

- $\mathcal{T}$ is the disjoint union of $\beta(mt_i)$ where $mt_i \in \mathbf{mt}$. More colloquially every triple is contained in exactly one model instance.

The following constraints define the inheritance, domain, range and cardinality constraints that the meta model instantiation must satisfy:

- if the object type $o_1^t \in \mathbf{O}_j^T$ is a subtype of the object type $o_2^t \in \mathbf{O}_j^T$ $(o_1^t \preceq o_2^t)$ then we have
$$\mu_{\mathbf{O}}(o_1^t, \mathbf{MT}_j) \subseteq \mu_{\mathbf{O}}(o_2^t, \mathbf{MT}_j). \tag{21}$$
- Sibling object types are disjoint. More specifically if $o_1^t, o_2^t, o_3^t \in \mathbf{O}_j^T$ are object types such that
$$o_2^t \preceq o_1^t, o_3^t \preceq o_1^t, o_2^t \not\preceq o_3^t, o_3^t \not\preceq o_2^t \tag{22}$$
then
$$\mu_{\mathbf{O}}(o_2^t, \mathbf{MT}_j) \cap \mu_{\mathbf{O}}(o_3^t, \mathbf{MT}_j) = \emptyset. \tag{23}$$
- If the value y of a statement is an object, i.e. there is a mapping from an object type to an object for a concrete model type $\mathbf{MT}_j$, then the pair of an

object type and a model type have to be part of the range definition in the meta model:

$$(x\ a\ y) \in \mathcal{T} \wedge y \in \mathbf{O} \Rightarrow \exists o^t, MT_j, (y \in \mu_{\mathbf{O}}(o^t, MT_j) \wedge (o^t, MT_j) \in range(a)) \tag{24}$$

The second equation further defines that if y points to an object, then there must exist an object type o^t and a model type MT_j that are part of an $\mu_{\mathbf{O}}$ mapping for y.

- If the value y of a statement is a data object then there must exist a datatype that is part of the range definition of the attribute in the meta model and there must be a mapping between the data type and the data object:

$$(x\ a\ y) \in \mathcal{T} \wedge y \in \mathbf{D} \Rightarrow \exists d^t \in \mathbf{D}^T\ d^t \in \mathrm{range}(a) \wedge y \in \mu_{\mathbf{D}}(d^t) \tag{25}$$

- If the value y of a statement is a model instance $\mathbf{mt}$, then a model type $\mathbf{MT}_j$ must be part of the range definition and the y value must correspond to the mapping of that model type to the model instance:

$$(x\ a\ y) \in \mathcal{T} \wedge y \in \mathbf{mt} \Rightarrow \exists \mathbf{MT}_j \in \mathrm{range}(a), y \in \mu_{\mathbf{mt}}(\mathbf{MT}_j) \tag{26}$$

- For each statement the attribute a of that statement must be part of the same model type from which the object x has been mapped:

$$(x\ a\ y) \in \mathcal{T} \Rightarrow \exists j\ a \in \mathbf{A}_j \wedge \exists o^t \in \mathrm{domain}(a), x \in \mu_{\mathbf{O}}(o^t, \mathbf{MT}_j) \tag{27}$$

- If the value y of a statement is a data object then the data type must be part of the same model type as the attribute:

$$(x\ a\ y) \in \mathcal{T}, a \in \mathbf{A}_i, y \in \mathbf{D} \Rightarrow \exists d^t \in \mathbf{D}_i^T, y \in \mu_{\mathbf{D}}(d^t) \tag{28}$$

- And for the cardinality constraints: if $x \in \mu_{\mathbf{O}}(o^t, MT_j)$, $a \in \mathbf{A}_i$ where $\langle m, n\rangle = card(o^t, a)$ then $m \leq |\{y\ :\ (x\ a\ y) \in \mathcal{T} \wedge y \in (\mathbf{O} \cup \mathbf{D} \cup \mathbf{mt})\}| \leq n$.

4 Application of FDMM to the 4R Modeling Language

To illustrate the usage of the FDMM formalism we will show in the following how the modeling language from the running example in Sect. 2.3 and instances of this modeling language can be formally described. We start by defining the model types that will represent the *4R portfolio models* and *4R business process models* by $\mathbf{MT}_{PO}$ and $\mathbf{MT}_{BP}$:

$$\mathbf{MT}_{\mathrm{PO}} = \left\langle \mathbf{O}_{\mathrm{PO}}^T, \mathbf{D}_{\mathrm{PO}}^T, \mathbf{A}_{\mathrm{PO}} \right\rangle, \mathbf{MT}_{\mathrm{BP}} = \left\langle \mathbf{O}_{\mathrm{BP}}^T, \mathbf{D}_{\mathrm{BP}}^T, \mathbf{A}_{\mathrm{BP}} \right\rangle \tag{29}$$

Next, we detail the sets of object types $\mathbf{O}_{\mathrm{PO}}^T$ and $\mathbf{O}_{\mathrm{BP}}^T$ for expressing what corresponds to the classes and relationclasses in ADOxx by:

$$\begin{aligned} \mathbf{O}_{PO}^T &= \{Business\text{-}transaction, relates\text{-}business\text{-}transaction\} \\ \mathbf{O}_{BP}^T &= \{FlowObject, Start, Decision, Activity, Parallelity, Join, End, \\ &\quad 4R\text{-}event, 4R\text{-}risk\text{-}aggregation, subsequent, aggregation \\ &\quad influences\} \end{aligned} \tag{30}$$

Thereby, the object type *FlowObject* is defined as an abstract type that has to be instantiated through one of its sub types. In addition the following subtype relationships hold between the following object types:

$$\begin{aligned}&Start \preceq FlowObject, Decision \preceq FlowObject, Activity \preceq FlowObject,\\&\quad Parallelity \preceq FlowObject, Join \preceq FlowObject, End \preceq FlowObject \qquad (31)\end{aligned}$$

The same is applied for detailing the sets of data types $\mathbf{D}^T_{PO}$ and $\mathbf{D}^T_{BP}$. Thereby, the $\mathbf{Enum}_{view}$ and $\mathbf{Enum}_{influence}$ types are used to represent the ADOxx enumeration attribute types with pre-defined values:

$$\begin{aligned}\mathbf{D}^T_{PO} &= \{String, Float, \mathbf{Enum}_{view}\}\\ \mathbf{Enum}_{view} &= \{Aggregated, Single\}\\ \mathbf{D}^T_{BP} &= \{String, Float, \mathbf{Enum}_{influence}\}\\ \mathbf{Enum}_{influence} &= \{Time\text{-}influence, Cost\text{-}influence,\\ &\quad Return\text{-}influence, Quality\text{-}influence\} \qquad (32)\end{aligned}$$

We continue by detailing the sets of attributes $\mathbf{A}_{PO}$ and $\mathbf{A}_{BP}$:

$$\begin{aligned}\mathbf{A}_{PO} &= \{ID, W, RE, RI, WE, \rho, relates\text{-}from, relates\text{-}to, Process, View\}\\ \mathbf{A}_{BP} &= \{Name, \epsilon, I, Time, Cost, Return, Quality, Influence\text{-}type,\\ &\quad subsequent\text{-}from, subsequent\text{-}to, aggregation\text{-}from,\\ &\quad aggregation\text{-}to, influences\text{-}from, influences\text{-}to\} \qquad (33)\end{aligned}$$

For attaching the attributes to the object types and defining their value range, we add according domain and range definitions. This can be done for example by attaching the *Name* attribute to the required object type and then defining its range to be of the data type *String*. By using the *FlowObject* abstract type we can do this for all object types that are defined as its subtypes:

$$\begin{aligned}\mathrm{domain}(Name) &= \{FlowObject, 4R\text{-}risk\text{-}aggregation, 4R\text{-}event\}\\ \mathrm{range}(Name) &= \{String\} \qquad (34)\end{aligned}$$

We then add the cardinality definitions for each of the object types and their attributes as shown here exemplarily for the name attribute:

$$\begin{aligned}&\mathrm{card}(FlowObject, Name) = \langle 1,1\rangle, \mathrm{card}(4R\text{-}risk\text{-}aggregation, Name) = \langle 1,1\rangle,\\ &\mathrm{card}(4R\text{-}event, Name) = \langle 1,1\rangle \qquad (35)\end{aligned}$$

As it has been done for the attributes of the data type *String* we can similarly define the domain, range, and cardinality functions for an attribute of the type *Float*. As already mentioned above, the FDMM formalism does not further specify the data types so that we would have for example:

$$\begin{aligned}&\mathrm{domain}(W) = \{Business\text{-}transaction\}, \mathrm{range}(W) = \{Float\}\\ &\mathrm{card}(Business\text{-}transaction, W) = \langle 0,1\rangle \qquad (36)\end{aligned}$$

In the same way, ADOxx attributes with pre-defined values can be represented in FDMM as shown in the following by inserting the data type set $\mathbf{Enum}_{view}$ in the range definition to specify the type of view that is used for a business transaction:

$$\text{domain}(View) = \{Business\text{-}transaction\}, \text{range}(View) = \{\mathbf{Enum}_{view}\}$$
$$\text{card}(Business\text{-}transaction, View) = \langle 1, 1\rangle \tag{37}$$

To permit references from one object to another model instance, e.g. to reference business transactions to corresponding 4R business process models, the following domain and range definitions are needed:

$$\text{domain}(Process) = \{Business\text{-}transaction\}, \text{range}(Process) = \{\mathbf{MT}_{\text{BP}}\}$$
$$\text{card}(Business\text{-}transaction, Process) = \langle 0, 1\rangle \tag{38}$$

Finally, we also give an example for defininig the equivalent of a relationclass based on an object type that connects to two other object types via "to" and "from" attributes:

$$\begin{aligned}
\text{domain}(influences\text{-}from) &= \{influences\}\\
\text{range}(influences\text{-}from) &= \{(4R\text{-}risk\text{-}aggregation, \mathbf{MT}_{\text{BP}})\}\\
\text{card}(influences, influences\text{-}from) &= \langle 1, 1\rangle\\
\text{domain}(influences\text{-}to) &= \{influences\}\\
\text{range}(influences\text{-}to) &= \{(Activity, \mathbf{MT}_{\text{BP}})\}\\
\text{card}(influences, influences\text{-}to) &= \langle 1, 1\rangle
\end{aligned} \tag{39}$$

Also for such an object type, that corresponds to a relationclass, attributes can be added in the same way as shown above:

$$\begin{aligned}
\text{domain}(Influence\text{-}type) &= \{influences\}\\
\text{range}(Influence\text{-}type) &= \{\mathbf{Enum}_{influence}\}\\
\text{card}(influences, Influence\text{-}type) &= \langle 1, 1\rangle
\end{aligned} \tag{40}$$

When defining relationclasses that can be used to connect multiple classes, the definition can be simplified by reverting to a supertype class as for example the *FlowObject* class and the *subsequent* relationclass:

$$\begin{aligned}
\text{domain}(subsequent\text{-}from) &= \{subsequent\}\\
\text{range}(subsequent\text{-}from) &= \{(FlowObject, \mathbf{MT}_{\text{BP}})\}\\
\text{card}(subsequent, subsequent\text{-}from) &= \langle 1, 1\rangle\\
\text{domain}(subsequent\text{-}to) &= \{subsequent\}\\
\text{range}(subsequent\text{-}to) &= \{(FlowObject, \mathbf{MT}_{\text{BP}})\}\\
\text{card}(subsequent, subsequent\text{-}to) &= \langle 1, 1\rangle
\end{aligned} \tag{41}$$

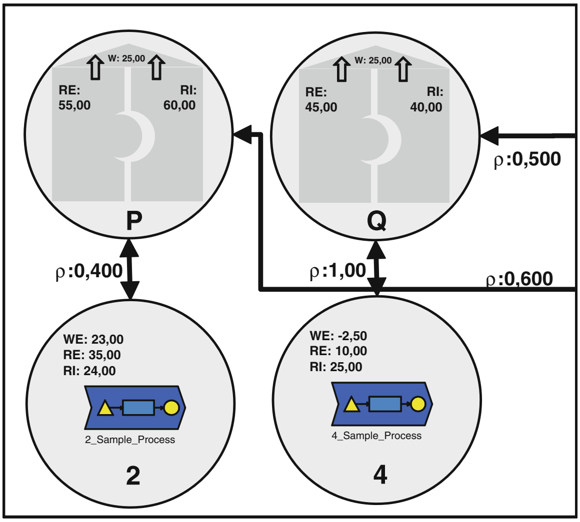


Fig. 3. Excerpt of a 4R Portfolio Model [13].

Based on these definitions for the model type we can describe the instantiation of a concrete model. As an example we use two models that have been described in [13] - see Figs. 3 and 4. They represent sample instances of a 4R portfolio model type and a 4R business process model type that shows how 4R events and 4R risk aggregations are used to represent the influence of risks on

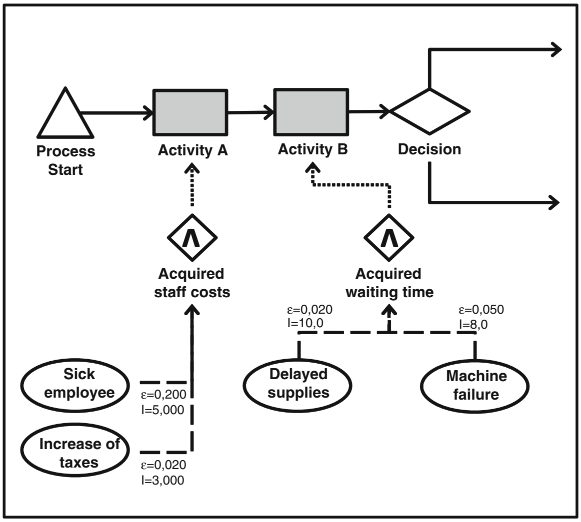


Fig. 4. Excerpt of a 4R Business Process Model [13].

the accomplishment of activities. We will use these models to describe some of its contents by using the FDMM formalism.

First we instantiate concrete models for the model types $\mathbf{MT}_{PO}$ and $\mathbf{MT}_{BP}$ based on a mapping from the meta model definition:

$$\mu_{\mathbf{MT}}(\mathbf{MT}_{PO}) = \{\mathbf{mt}_{po1}\}, \mu_{\mathbf{MT}}(\mathbf{MT}_{BP}) = \{\mathbf{mt}_{bp1}\} \tag{42}$$

Next, we instantiate objects based on mappings from the object types. We show this exemplarily for some instances of the business transaction, activity, 4R event, and 4R aggregation object types:

$$\begin{aligned}
\mu_{\mathbf{O}}(Business\text{-}transaction, \mathbf{MT}_{PO}) &= \{BT_P, BT_Q\} \\
\mu_{\mathbf{O}}(Activity, \mathbf{MT}_{BP}) &= \{Activity_A, Activity_B\} \\
\mu_{\mathbf{O}}(4R\text{-}event, \mathbf{MT}_{BP}) &= \{Machine\text{-}failure, Delayed\text{-}supplies\} \\
\mu_{\mathbf{O}}(4R\text{-}aggregation, \mathbf{MT}_{BP}) &= \{Acquired\text{-}waiting\text{-}time\}
\end{aligned} \tag{43}$$

Similarly we can instantiate object types that can later act as relations such as the influences type:

$$\mu_{\mathbf{O}}(influences, \mathbf{MT}_{BP}) = \{influences_1, influences_2\} \tag{44}$$

Subsequently, we also show the mappings of some data types to data objects in order to later assign them as values for attributes:

$$\begin{aligned}
\mu_{\mathbf{D}}(String) &= \{'ActivityA', 'ActivityB', 'DelayedSupplies', \\
&\quad 'Machinefailure', Acquired\text{-}waiting\text{-}time\} \\
\mu_{\mathbf{D}}(Float) &= \{25.00, 55.00, 60.00\} \\
\mu_{\mathbf{D}}(Time\text{-}influence) &= \{'time\text{-}influence'\}
\end{aligned} \tag{45}$$

And finally we can define the relationships between the objects and the data object by using the attributes in the form of triple statements, e.g. to express the names of concrete objects and the values of attributes and defining relations:

$$\begin{aligned}
&(Activity_A \;\; Name \;\; 'ActivityA') \in \beta(\mathbf{mt}_{bp1}), (BT_P \;\; W \;\; 25.00) \in \beta(\mathbf{mt}_{po1}), \\
&(influences_1 \;\; influences\text{-}from \;\; Acquired\text{-}waiting\text{-}time) \in \beta(\mathbf{mt}_{bp1}), \\
&(influences_1 \;\; influences\text{-}to \;\; Activity_B) \in \beta(\mathbf{mt}_{bp1})
\end{aligned} \tag{46}$$

And for detailing the type of influence by specifying the attribute value that is available based on a pre-defined data type:

$$(influences_1 \;\; Influence\text{-}type \;\; 'time\text{-}influence') \in \beta(\mathbf{mt}_{bp1}) \tag{47}$$

5 FDMM for Implementations on the ADOxx Platform

The ADOxx meta modeling approach as it has been described in Sect. 2.2 has been implemented in the form of an industry-ready, client-server based software platform. One core functionality of the platform is, that it permits to specify meta models in a proprietary definition language and automatically generates according model editors. Both meta models and models are thereby automatically stored in a relational database. In addition, the platform provides a number of extension functionalities such as a proprietary scripting language for defining mechanisms and algorithms on models, a generic query language for models, documentation functionalities for generating reports about models or a web service interface for establishing couplings to third party tools. These functionalities are complemented by a multi-user rights management for controlling the accessibility to the different platform components and the models.

In the following we will illustrate how the specifications using the FDMM formalism can be used to implement meta models on the ADOxx platform. We start by showing how the model type for 4R business process models as defined in Eq. 29 is translated into the ADOxx Library Language (ALL). These ALL definitions are then used by the ADOxx platform to create the meta model and automatically provide according model editors for the thus specified model types:

Example Definition of the 4R Business Process Model Type in ALL

```
BUSINESS PROCESS LIBRARY <4R-IEB-Library>
    ATTRIBUTE <Version number>
    VALUE ""
    ...
    ATTRIBUTE <Modi>
    VALUE "MODELTYPE \"4R Business process model\" from:none
    plural:\"4R Business process models\"
        INCL \"Start\"
        INCL \"Decision\"
        INCL \"Activity\"
        INCL \"Parallelity\"
        INCL \"Join\"
        INCL \"End\"
        INCL \"Subsequent\"
        INCL \"4R risk aggregation\"
        INCL \"4R event\"
        INCL \"aggregation\"
        INCL \"influences\"
        MODE \"All modelling objects\" from:all
        MODE \"Documentation\" from:all no-modeling"
   ...
```

The 4R business process model type is thereby defined to contain a number of *classes* that are not abstract and *relationclasses* and can therefore be instantiated to object instances in a model editor. When mapping the definitions in

FDMM to ALL, it has to be decided, which object types in FDMM become classes and which become relationclasses in ALL. As relationclasses in ALL can only represent binary relations between object types of the cardinality $< 1, 1 >$, only object types that satisfy this restriction can be mapped to a relationclass. For other relations between object types and between object types and model types, ALL offers the *interref* attribut type. As a rule of thumb, FDMM object types that correspond to edges in graph-like model types can be mapped to relationclasses.

The definition of classes in ALL is accomplished using the ‘CLASS’ keyword. As shown in the example code below, classes are arranged in a superclass/subclass hierarchy using a colon followed by the name of the super class. Thereby, every class in ALL has to be a direct or indirect subclass of the predefined “__BP-construct__” class. These hierarchy definitions correspond to the definition of Eq. 31 as shown for the FlowObject class and the Start class:

Example Definition of the Start Class in ALL

```
CLASS <FlowObject> : <__BP-construct__>
...
CLASS <Start> : <FlowObject>
...
```

For the classes also the corresponding attributes as defined in FDMM by the domain, range, and cardinality statements can be added in ALL. Thereby, the name attribute is automatically added by ADOxx as it constitutes the unique identifier of instances of a class. For every attribute in ALL, the type and potential additional information have to be specified. We show this in the following example for the definition of the Business transaction class as specified by the Eqs. 36, 37:

Example Definition of the Business Transaction Class and the W and View Attributes in ALL

```
CLASS <Business transaction> : <__BP-construct__>
...
ATTRIBUTE <W>
    TYPE DOUBLE
    VALUE 0
...
ATTRIBUTE <View>
    TYPE ENUMERATION
    FACET <EnumerationDomain>
    VALUE "Aggregated@Single"
...
```

As has already been mentioned above, relations between object types that do not satisfy the cardinality restrictions of relationclasses in ALL or that relate object types and model types, have to be specified as interref attributes in

ALL. Also, in case a model type should not contain visible relations between class instances, this attribute type can be chosen. We show this for the Process attribute of the business transaction model type that relates an instance of a Business Transaction class to a 4R business process model type as specified in Eq. 38:

Example Definition of the Process Interref Attribute in ALL

```
...
ATTRIBUTE <Process>
    TYPE INTERREF
        ...
        FACET <AttributeInterRefDomain>
        VALUE "REFDOMAIN MODREF
     mt:\"4R Business process model\"
     max:1"
...
```

The definition of relationclasses in ALL requires the definition of a start class, indicated by the FROM keyword, and a target class, indicated by the TO keyword. We show this by the example of the influences relationclass that is based on the specifications in Eq. 39 and the definition of the Influence type attribute in Eq. 40:

Example Definition of the Influences Relationclass in ALL

```
RELATIONCLASS <influences>
    FROM <4R risk aggregation>
    TO <Activity>
    ...
    ATTRIBUTE <Influence type>
    TYPE ENUMERATION
    FACET <EnumerationDomain>
    VALUE "Time influence@Cost influence@Return influence@
    Quality influence"
    ...
```

In order to permit the automatic generation of model editors from the ALL definitions, two more aspects have to be added. First, it has to be defined, which attributes of a class or a relationclass can be edited by a user in the model editor. This is defined via the AttrRep attribute as shown below for the example of the Influences relationclass:

Example Definition of the AttrRep Attribute of the Influences Relationclass in ALL

```
...
ATTRIBUTE <AttrRep>
    TYPE STRING
    VALUE "NOTEBOOK
    ATTR \"Influence type\""
...
```

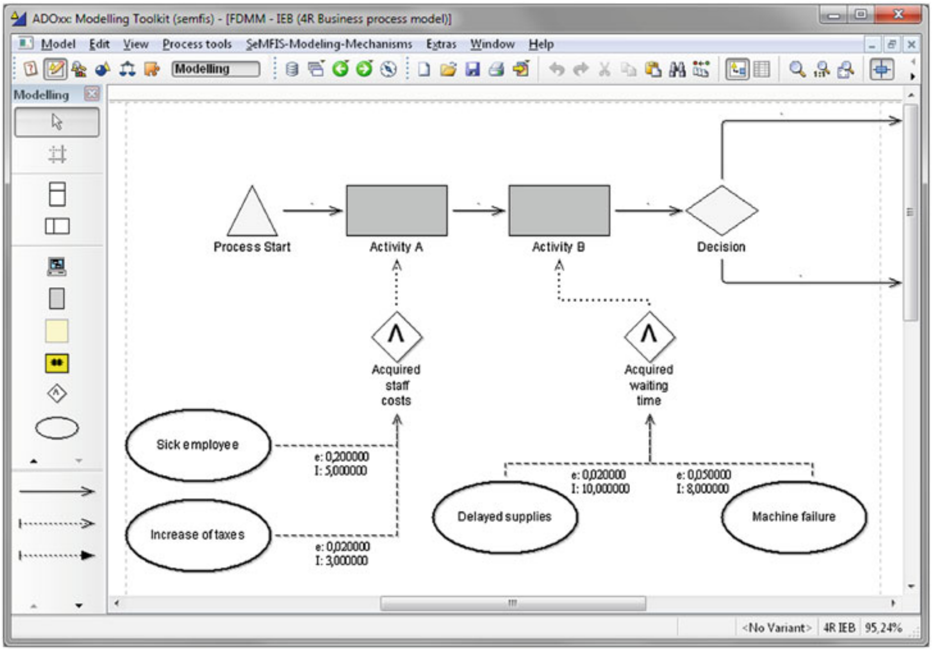


Fig. 5. Screesnhot of the Model Editor for 4R Business Process Models on ADOxx.

Second, also a graphical representation has to be added if the classes and relationclasses should be represented in a graphical way in the model editor. This is defined via the GraphRep class attribute that contains a proprietary grammar for defining the visual representation of a class or a relationclass. It can either be coded by hand or defined using a visual editor as provided by the Open Models Initiative[4]:

Example Definition of the GraphRep Classattribute of the 4R Event Class in ALL

```
CLASS <4R event> : <_4R-Superclass_>
...
    CLASSATTRIBUTE <GraphRep>
    VALUE "GRAPHREP
        PEN w:0.07cm
        ELLIPSE x:0cm y:0cm rx:2cm ry:1cm
        ATTR \"Name\" h:c w:c"
```

Finally, when all ALL definitions are in place, the ADOxx platform automatically generates model editors for the defined model types. As an example, an editor for the 4R business process models is shown in Fig. 5.

[4] See http://www.openmodels.at/web/omi/services

6 Discussion

Based on the constructs of the FDMM formalism it is possible to specify ADOxx meta models and models in a mathematically rigorous way that builds only upon set theory and first order logic statements. With these mathematical foundations, the ADOxx meta modeling approach can be compared to other approaches and used as a basis for further formal specifications or implementations. As has been discussed in [27], the FDMM formalism differs in several aspects from the ADOxx Library Language. In the paper at hand this could be shown in particular for the translation of object types into classes and relationclasses in ALL. As ALL requires the definition of relationclasses for the visualization of edges, not all object types specified in FDMM can be used for this purpose. Therefore, at this point it has to be decided by the developer of the ALL definition, how object types should be translated into ALL classes and relationclasses. In addition, also graphical representations and the definitions for editable attributes have to be added during this translation as they are not contained in FDMM. Regarding the graphical representations, it could thereby also reverted to the approach of semantic visualization that permits to automatically assign graphical representations based on semantic annotations [23].

7 Related Work

When comparing FDMM and ADOxx to similar approaches in the literature, two directions can be taken. The first is the comparison to other meta modeling approaches and the second is the comparison to other kinds of formalizations for meta modeling approaches.

Based on the classification proposed by [20], the FDMM and the ADOxx meta modeling approach directly compare to *domain-specific modeling approaches* that view meta models as language specifications. This is in contrast to approaches that treat meta models as *software structure specifications*, which is the typical use case for approaches such as EMOF [28], EMF [29] or KM3 [30]. A common aspect of domain-specific modeling approaches is the creation of visual model editors from meta models that are based on one pre-defined meta meta model and that use a graphical representation for the concrete syntax of the defined language. [8] also denote this direction as *heavyweight approaches* of language definition and distinguish it from *lightweight approaches* that adapt a generic meta model with domain-specific concepts. An example for the latter direction would be the use of the profile package in UML, e.g. to extend existing meta classes with the stereotyping mechanism [31].

FDMM and ADOxx can be directly compared to the approaches analyzed in [8]: thereby a core feature of ADOxx and FDMM that is shared with the GME and ARIS meta modeling approaches is the use of model types for defining the grouping of object types and their instances. In contrast to all approaches compared by Kern et al. and ADOxx, FDMM does not use any relation concept as a first class concept. Neither ADOxx nor FDMM use explicit *role type* concepts that provide further mechanisms for specifying relationships such as semantic dependencies between object types. However, in ADOxx such concepts can be expressed using the ADO*script* language and enforced during modeling.

For the meta modeling approaches mentioned above, formalizations have been discussed for EMOF e.g. [32,33] and KM3 [30]. However, they differ from ADOxx in regard to their focus on the specification of software structures. Another approach that shows some similarities to the way the FDMM formalism has been conceived can also be found in the specification of the Object Constraint Language (OCL) [34][Annex A]. However, the main difference is that FDMM is directed towards supporting the representation of meta models and models. The OCL specification does not describe a meta modeling approach but rather an approach to formalize one particular modeling language, i.e. UML together with constraints.

Furthermore, the domain, range, and card functions and the associated constraints described for them have a similarity with the notions of domain, range and cardinality restrictions used in description logics [35]. In contrast to the description logic case, our work is not intended to give a semantics for some formal language. Instead it is intended to provide a formal description of an existing system that has been effectively used in several application domains.

8 Conclusions and Outlook

In this paper we presented a formalism to describe the core constituents of the ADOxx meta modeling approach and showed its application to a concrete modeling language as well as how it can act as a basis for an implementation. It is the first formal definition for ADOxx meta modeling concepts and is therefore expected to be of benefit also for other projects using the ADOxx approach. Future work will therefore include the application to further modeling languages and the evaluation of the usability of the formalism. This concerns in particular the definition of algorithms, e.g. for describing analyses and simulations of models. Finally, it will also be investigated how the formalism can be represented visually to enhance the interaction with it and enable the easy re-use of formal meta model and model statements.

Acknowledgement. Parts of the work on this paper have been funded by the Austrian Science Fund (FWF) in the course of an Erwin-Schrödinger fellowship project number J3028-N23.

References

1. Kaschek, R.: On the evolution of conceptual modeling. In: Dagstuhl Seminar Proceedings, vol. 08181 (2008)
2. Karagiannis, D., Fill, H.G., Höfferer, P., Nemetz, M.: Metamodeling: Some application areas in information systems. In: Kaschek, R., Kop, C., Steinberger, C., Fliedl, G. (eds.) UNISCON 2008. LNBIP, vol. 5, pp. 175–188. Springer, Heidelberg (2008)

3. Wand, Y., Weber, R.: Research commentary: information systems and conceptual modeling - a research agenda. Inf. Syst. Res. **13**(4), 363–376 (2002)
4. Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.): Conceptual Modeling: Foundations and Applications. Springer, Heidelberg (2009)
5. Koch, S., Strecker, S., Frank, U.: Conceptual Modelling as a New Entry in the Bazaar: The Open Model Approach. In: Damiani, E., Fitzgerald, B., Scacchi, W., Scotto, M., Succi, G. (eds.) Open Source Systems. IFIP, vol. 203, pp. 9–20. Springer, Boston (2006)
6. Karagiannis, D., Grossmann, W., Hoefferer, P.: Open model initiative - a feasibility study. http://cms.dke.univie.ac.at/uploads/media/Open_Models_Feasibility_Study_SEPT_2008.pdf (2008). Accessed 04 April 2010
7. Hinkelmann, K., Nikles, S., Wache, H., Wolff, D.: An enterprise architecture framework to organize model repositories. In: Woitsch, R., Micsik, A. (eds.) OKM Open Knowledge Models, Workshop W3 at EKAW 2010 (2010)
8. Kern, H., Hummel, A., Kuehne, S.: Towards a comparative analysis of meta-metamodels. In: The 11th Workshop on Domain-Specific Modeling, Portland, USA. http://www.dsmforum.org/events/DSM11/Papers/kern.pdf (2011). Accessed 05 Jan 2012
9. Harel, D., Rumpe, B.: Meaningful modeling: What's the semantics of "semantics"? IEEE Comput. **37**(10), 64–72 (2004)
10. 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). Full version available at: http://www.dke.univie.ac.at/mmp/FullVersion_MMP_DexaECWeb2002.pdf
11. Kuehn, H.: The ADOxx Metamodelling Platform. In: Workshop on Methods as Plug-Ins for Meta-Modelling. Klagenfurt, Austria http://www.openmodel.at/c/document_library/get_file?uuid=7516b7c5-a525-4d92-929e-6c11e5da9d39&groupId=10122 (2010)
12. Bork, D., Sinz, E.: Design of a SOM Business Process Modelling Tool based on the ADOxx meta-modelling Platform. In: De Lara, J., Varro, D. (eds.) Proceedings of the Fourth International Workshop on Graph-Based Tools, EASST (2010)
13. Fill, H.G., Gericke, A., Karagiannis, D., Winter, R.: Modellierung fuer Integrated Enterprise Balancing (German: Modeling for Integrated Enterprise Balancing). Wirtschaftsinformatik **06**, 419–429 (2007)
14. Schwab, M., Karagiannis, D., Bergmayr, A.: i* on ADOxx(R): A Case Study. In: Proceedings of the 4th International i* Workshop - iStar10 - CAiSE Workshop Proceedings, pp. 92–97. Springer (2010)
15. Nemetz, M.: A meta-model for intellectual capital reporting. In: Reimer, U., Karagiannis, D. (eds.) PAKM 2006. LNCS (LNAI), vol. 4333, pp. 213–223. Springer, Heidelberg (2006)
16. Hofer, S.: Instances over algorithms: A different approach to business process modeling. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) PoEM 2011. LNBIP, vol. 92, pp. 25–37. Springer, Heidelberg (2011)
17. Braun, C., Winter, R.: A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. In: Desel, J., Frank, U. (eds.) Enterprise Modelling and Information Systems Architectures, pp. 64–79. Gesellschaft fuer Informatik, Bonn (2005)
18. Demirkan, H., Kauffman, R.J., Vayghan, J.A., Fill, H.G., Karagiannis, D., Maglio, P.: Service-oriented technology and management: Perspectives on research and practice for the coming decade. Electron. Comm. Res. Appl. **7**(4), 356–376 (2008)

19. Harel, D., Rumpe, B.: Modeling languages: Syntax, semantics and all that stuff - part i: The basic stuff. Technical Report MCS00-16, The Weizmann Institute of Science, 22 August 2000
20. Sprinkle, J., Rumpe, B., Vangheluwe, H., Karsai, G.: Metamodelling - state of the art and research challenges. In: Giese, H., Karsai, G., Lee, E., Rumpe, B., Schätz, B. (eds.) MBEERTS. LNCS, vol. 6100, pp. 57–76. Springer, Heidelberg (2010)
21. Harmon, P.: The BPTrends 2010 BPM Software Tools Report on BOC's Adonis Version 4.0 http://www.bptrends.com/publicationfiles/2010%20BPM%20Tools%20Report-BOCph.pdf (2010). Accessed 10 Oct 2011
22. Junginger, S., Kuehn, H., Strobl, R., Karagiannis, D.: Ein Geschaeftsprozessmanagement-Werkzeug der naechsten Generation - ADONIS: Konzeption und Anwendungen (German: ADONIS: A next generation business process management tool - Concepts and Applications). Wirtschaftsinformatik **42**(5), 392–401 (2000)
23. Fill, H.G.: Visualisation for Semantic Information Systems. Gabler, Wiesbaden (2009)
24. Faisst, U., Buhl, H.: Integrated Enterprise Balancing mit integrierten Ertrags- und Risikodatenbanken (German: Integrated Enterprise Balancing with integrated Return and Risk Databases). Wirtschaftsinformatik **47**(6), 403–412 (2005)
25. Gericke, A., Fill, H.G., Karagiannis, D., Winter, R.: Situational Method Engineering for Governance. Risk and Compliance Information Systems. In: DESRIST. ACM (2009)
26. Karagiannis, D., Junginger, S., Strobl, R.: Introduction to Business Process Management Systems Concepts. In: Scholz-Reiter, B., Stickel, E. (eds.) Business Process Modelling, pp. 81–106. Springer, Berlin (1996)
27. Fill, H.G., Redmond, T., Karagiannis, D.: FDMM: A Formalism for Describing ADOxx Meta Models and Models. In: Maciaszek, L., Cuzzocrea, A., Cordeiro, J. (eds.) Proceedings of ICEIS 2012, Wroclaw, Poland, vol 3, pp. 133–144. (2012)
28. Object Management Group: Omg meta object facility (mof) core specification version 2.4.1. http://www.omg.org/spec/MOF/2.4.1/PDF/ (2011)
29. McNeill, K.: Metamodeling with EMF: Generating concrete, reusable Java snippets. http://www.ibm.com/developerworks/library/os-eclipse-emfmetamodel/index.html?S_TACT=105AGX44&S_CMP=EDU (2008)
30. Jouault, F., Bézivin, J.: KM3: a DSL for Metamodel Specification. In: Gorrieri, R., Wehrheim, H. (eds.) FMOODS 2006. LNCS, vol. 4037, pp. 171–185. Springer, Heidelberg (2006)
31. Omg, O.M.G.: Unified modeling language (uml) specification: Infrastructure version 2.0. Technical report. http://www.omg.org/docs/ptc/04-10-14.pdf (2004). Accessed 12 Mar 2006
32. Poernomo, I.: The Meta-Object Facility Typed. In: SAC'06, Dijon, France, pp. 1845–1849. ACM (2006)
33. Favre, L.M.: Formalization of MOF-Based Metamodels. In: Favre, L.M. (ed.) Model Driven Architecture for Reverse Engineering Technologies. Information Resources Management Association (2010)
34. OMG: Object constraint language specification version 2.2. Technical report. http://www.omg.org/spec/OCL/2.2 (2010). Accessed 11 Jan 2012
35. Baader, F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)

The Enterprise Architecture Body of Knowledge as an Evolving Discipline

Hadi Kandjani and Peter Bernus(✉)

Centre for Enterprise Architecture Research and Management (CEARM),
School of ICT, Griffith University, Brisbane, Australia
{H.Kandjani, P.Bernus}@griffith.edu.au

Abstract. Enterprise Architecture (EA) as an area of interdisciplinary study relies on models, methods and theories of many disciplines. The article explores the linkage between the needs of enterprise problem domains, the evolution of domain specific disciplines, and the EA body of knowledge. A cybernetic view is presented in an attempt to explain the effects of an important driver of discipline development, namely the change in the complexity of application domains. For the EA discipline (EAD), as any other developing discipline, there should exist a commonly accepted terminology, allowing interdisciplinary theories to be stated, which in turn facilitate the creation of cross disciplinary models and methodologies. While there already exists a fundamental and generalised theory of EA, GERAM, it is a minimalist theory, not prescribing any particular reference models or any concrete methodology, thus there is a constant need to relate domain specific results to the generalised theory, whereupon the evolution of one needs to have impact on the other. In this article we treat the discipline-as-a-system, and use Beer's Viable System Model (VSM) to discuss three basic components of EAD as a viable system. A 'co-evolution mechanisms' for EAD is proposed, and a cybernetic model of co-evolution applied to EAD. We also discuss a cybernetic model of EAD using Checkland's model for discipline development.

Keywords: Enterprise architecture discipline · Unified theory · Viable system model · Co-evolution path model · Enterprise architecture cybernetics

1 Introduction

For Enterprise Architecture (EA), like any other developing and evolving discipline, there should exist a theory, with terminology and rules capable of unifying the constituent models and methodologies developed by contributing disciplines. There already exists a fundamental theory of EA: GERAM, however it is a minimalist theory, not prescribing any particular reference models or methodology [1, 2].

The GERAM framework includes a terminology, with the concepts of enterprise-entity, life cycle, life history, modelling languages, models, instantiation and tools, and rules of life cycle relationships, conceived for use in the management of any change. The framework provides the user with a generic reference model for

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 452–470, 2013.
DOI: 10.1007/978-3-642-40654-6_27,

constituents of a life cycle and a modelling framework, with complete subdivision of view point concepts (NB these are called 'views' in the original GERAM document, but for compatibility with ISO 42010 [3] we use here the term viewpoint). Note that this architecture framework and its theory are independent from the domain and type of change.

For pragmatic purposes, practitioners who develop(ed) particular architectural frameworks add methodologies and/or reference models specific to the application domain and/or the type of change for which they are intended to be applicable.

For example, DoDAF [4], as a particular architectural framework, includes an interoperability capability assessment reference model, defined as a series of levels, called 'Levels of Information System Interoperability' (LISI).

TOGAF [5], as a particular architecture framework, comes with a methodology to develop IT architecture, called the 'Architecture Development Method' (ADM).

These particular frameworks (DoDAF, TOGAF etc.) are domain dependent and were developed for a specific type of change; whereupon, general architectural frameworks, such as GERAM, are independent from the domain and type of change.

The critical question in this paper is that: how is it possible to extend the EA Body of Knowledge with common elements that are domain independent as well as independent from the type of change? In other words: what is a unified evolving model of EA Body of Knowledge? By answering this question, we could in fact have an extension of the theory to the architecture of any large scale complex system.

Cybernetics and General Systems Theory (GST) have previously attacked these types of problems at the same, or similar, level of abstraction and generality. Therefore, to develop and extend the EA discipline we need to incorporate the apport of previously related disciplines and their theories into a unified theory. As this will no doubt be a long term process we must treat EA as an evolving and developing discipline.

Norbert Wiener defined cybernetics as "the science of control and communication in the animal and machine" [6]. Ashby [7] also calls cybernetics the art of "steermanship" which studies co-ordination, regulation and control of systems, arguing that the "truths of cybernetics are not conditional on their being derived from some other branch of science". Therefore the field embraces a set of self-contained groundings and foundations, which Ashby tried to describe in his book (ibid). He addressed the complexity of a system as one of the peculiarities of cybernetics and indicated that cybernetics prescribes a scientific method of dealing with complexity as a critical attribute of a system.

Stafford Beer believed that the dynamics of enterprises is about "the manipulation of men, material, machinery and money: the four Ms", plus an even more fundamental "manipulation" (from microscopic biological organisms to large scale systems, including enterprises): the "management of complexity" [8, 9].

Enterprises are best understood as intrinsically complex adaptive living systems: they can not purely be considered as 'designed systems', as deliberate design/control episodes and processes ('enterprise engineering' using design models) are intermixed with emergent change episodes and processes (that may perhaps be explained by models). The mix of deliberate and emerging processes can create a situation in which the enterprise as a system is in a dynamic equilibrium (for some stretch of time) – a property studied in General Systems Theory [10, 11]. The evolution of the enterprise (or enterprises, networks, industries, the economy, society, etc.) includes emergent as well as the deliberate aspects of system change, therefore an EA theory must interpret previous research in both.

This unified theory is indeed to be a developing theory, describing evolution of the EA body of knowledge, therefore it should remain open for further continuous contributions of EA practitioners and researchers. The integrating, or interdisciplinary, aspect of EA manifests when studying enterprises as complex systems. Here, researchers not only apply models, methods and theories of management and control (and apply the same from engineering, linguistics, cognitive science, environmental science, biology, social science, artificial intelligence, systems thinking and cybernetics), there needs to be a synthesis of these.

Given this standpoint many theoreticians can contribute to the development of a unified theory of designing/architecting complex systems, taking into account a list of concerns expressed (issues addressed) by different disciplines that are related to 'designing' systems. We call these design- or architecture- concerns 'metaphors'. We can describe the architecture (i.e. 'architecting') process as:

- a *Conversation* between the controller of the system, the system's 'operations' and the controllers of environmental 'entities' (Conversation Theory [12]),
- a *Decisional & Resource Allocation Process* (using GRAI Grid [13, 14]),
- a *Complex Process* managed to reduce complexity and improve the likelihood of success (applying Axiomatic Design Theory [15–17],
- an *Emergent and Evolutionary Process* (using Complex Adaptive Systems Theory [18, 19],
- a *Planning & Prediction Process* (using Multi-agent Systems Theory theories [20, 21],
- a *Participatory Process* (using models of Participatory Design [22, 23],
- a *Change Process* (using Re-engineering Methods and approaches [24], and
- a *Learning Process* (using Systems thinking and Cybernetics theories [7, 25–27].

To develop a unified theory of large scale systems evolution as an extension to the EA body of knowledge, one should therefore review previously developed theories, models and terminologies that study the problem of designing/architecting complex systems.

In order to ensure that this unified theory has sufficient breadth and depth, it is useful to analyse how researchers previously considered the problems and concerns of this area. This useful to understand underlying concerns and problems that various researchers have had when designing architectural processes/frameworks, but it could also bring new discoveries via this theory unification process.

When studying enterprises as (partially designed and partially evolving) complex evolving systems, many researchers and practitioners implicitly apply methods and models derived from laws and theories of systems thinking and cybernetics. Cybernetics, as an interdisciplinary movement, has formulated multiple laws and theories of complex systems, but each one is presented on a different level of formality, generality and abstraction. Consequently, the application of these laws and theories in Enterprise Architecture (EA) also lack harmony. Therefore, we introduce EA Cybernetics as a field of EA with the intention to harmonise, formalise, synthesise and systematise results of multiple disciplines, using systems thinking and cybernetics, for a concerted and coherent application.

EA Cybernetics is the re-interpretation of old- and new theories to understand their individual contributions, and to point at the need for genuinely new results when designing and creating complex systems. Cybernetic thinking is a way to unify/relate the apport of multiple disciplines as explained in the 'architecting concerns'. Such a synthesis would be the source of a new, unified theory of EA, giving rise to more powerful theories, methodologies and reference models than available today.

2 Viability of the EA Discipline and Effective EA Practice

In this section we propose a viable model of EA as an interdisciplinary discipline of designing, creating and maintaining complex systems.

2.1 Beer's Viable System Model

Beer [28] describes every system as consisting of three main interacting components: Management, Operation and Environment (see Fig. 1).

Every system of interest (circle in the figure) has a meta-system as its management (represented as a square in the figure) and operates in an environment (represented as an oval shape in the figure), where each component could be further decomposed into more detailed elements. There are communication channels among these three components to keep the operation in homeostasis: these channels are called 'variety attenuators' and 'variety amplifiers' [9, 28, 29].

According to Beer [28] the 'variety' of the operations is always less than that of the environment, and the 'variety of management' is always less than the variety of

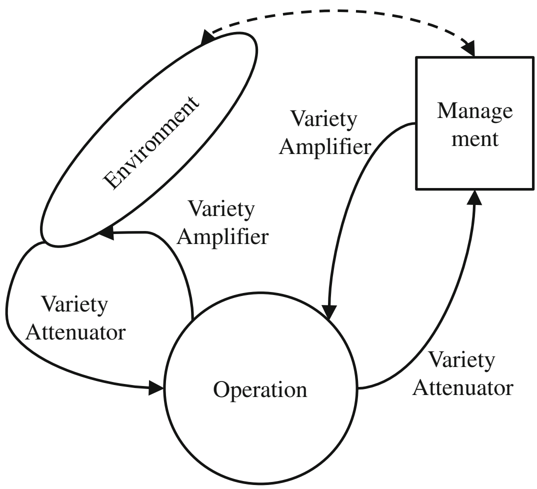


Fig. 1. Three components of the Viable System Model (VSM) [9].

operations. In contrast, based on Ashby's law of requisite variety [7], in order to achieve dynamic stability under change, the variety of operations should be equal to that of its relevant environment, and the variety of management should be at least equal to that of operations. In fact variety attenuation and amplification mechanisms need to be designed in order to keep the system of interest viable ('evolvable') in its environment.

However, sometimes the enterprise's mission, instead of viability and 'eternality' of the enterprise, is a temporary existence, therefore the demolition or deconstruction of enterprise or enterprise entities is an equally important aspect to consider. Like in construction and civil engineering, a demolition plan is a set of processes to tear down buildings or other structures. The same concept applies in EA.

For this purpose, a cybernetic model of EAD must cover the complete life history of the enterprises as systems (and system of systems), including all significant life events – creation, reproduction, merger, as well as decommissioning/end of life. It is especially this last one, end of life, that has received insufficient attention in literature, therefore we propose the concept of Fatal System Model (FSM) stressing that EA should address the cradle-to-grave aspect of the enterprise, from birth (creation of enterprises or agglomerations thereof) to decommissioning (states which finally make the enterprise collapse or dissolve, perhaps preserving valuable elements for re-use).

2.2 The EA Discipline as a Viable System

It is possible to map the three components of Beer's VSM to the EA Discipline itself, and to its surrounding environment. We consider the enterprise-related disciplines as 'operation' shown as a circle, and the EA discipline as its integrating and interdisciplinary meta-system ('management') shown as a square, with EA's task being to observe and cross-fertilise enterprise problem domains as well as to observe the 'environment' (Fig. 2).

There are communication channels (acting as variety attenuators and amplifiers) among the three components to achieve/maintain the requisite variety, i.e. in order to keep the EA discipline and its related disciplines as a system in homeostasis.

The EA discipline acts as a meta-system that investigates the enterprise problem domains and using attenuation mechanism, invokes the relevant terminology, models and theories from enterprise-related disciplines (e.g. systems thinking and cybernetics, industrial engineering, management science, control engineering, information and communication technology) to respond to new issues arising in enterprise problem domains.

Changes in the problem domains mandate the evolution of individual enterprise related disciplines, so as to respond to the new requirements of the evolving environment. In fact the evolution of enterprise-related disciplines and enterprise problem domains are coupled and mutually dependent, and the EA Discipline should act as a meta-system/management system regulating the requisite variety between operations and the environment (Fig. 3).

In order to harmonise this co-evolution, we need to understand what are the relevant mechanisms to guarantee an effective evolution of EA itself.

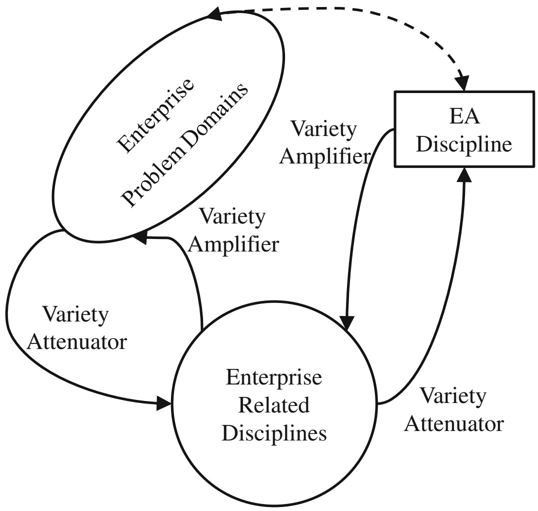


Fig. 2. Three components of a Viable EA Discipline.

If we consider EA as 'problem solving', then the step by step stages of co-evolution would be: (1) diagnose a significant problem in the enterprise problem domain, (2) invoke one or more relevant disciplines studying the enterprise problem domain and decide if such multi-disciplinary combined action is adequate, and if not, then (3) provide solutions for enterprise problems by harmonising and integrating multiple theories, models, techniques and methods from relevant disciplines in a synthesis (new or extended theory), and (4) adopt any 'new' case records of relevant disciplines and mutual contributions of EA and relevant disciplines into the EA Body of Knowledge.

The need for a unifying theory clarifies the role of EA as a meta-system (as in Beer's VSM) that answers the question: (a) what enterprise problems domains would be (or should be) addressed in a specific EA practices?, (b) what would be the invoked disciplines targeting the problem domains to solve the problem in combined use?, (c) how to formalise and harmonise other disciplines' contributions and apply them in an EA practice?

As the invoked disciples are continuously progressing and evolving in their specific domain and field of application, a more effective EA practice could be guaranteed if the evolution of these disciplines were influences or monitored by the EA discipline and the findings reflected in EA theory and practice when necessary.

3 Co-evolution Mechanisms for an Evolving EA Discipline

We discussed three components of a viable EA discipline in Sect. 2, now the question arises: what are the mechanisms to keep the requisite variety of the EA discipline as a viable system?

3.1 Co-evolution Path Model, Dynamic Homeostasis Vs. Dynamic Hetereostasis

Beer [8] argues that a key property of a viable system and a "measure of its submission to the control mechanism" is its ability to maintain its equilibrium or homeostasis, which he defines as "constancy of some critical variables (outputs)". In our model of co-evolution, we define the dynamic sustenance of requisite variety based on Ashby's law: "only variety can destroy variety" [7], paraphrased by Beer [28] as "variety absorbs variety".

Here, 'variety' is the number of possible states of a system [29], or as recently re-interpreted and refined by Kandjani and Bernus [30], the number of relevant states of a system.

For a system to dynamically achieve/maintain requisite variety and to be in dynamic equilibrium, the system requires communication channels and feedback loops. These channels serve as self-perpetuating mechanism and include both attenuation and amplification mechanisms. (Note that for the discussion below what we call a 'system' includes the system's controller).

Considering the system and its environment as two coupled entities, if one component is perturbed, the effect of that perturbation on the other component is either amplified through positive feedback, or may be reversed (attenuated) through negative feedback.

The role of the negative feedback loop is to reverse the effect of the initial perturbation and restore the system's homeostasis (in which critical variables are stable), while positive feedback can create unstable states [31].

We observe that both a system and its environment (including systems in that environment) evolve, and the change can create imbalance between the requisite variety (maintained by the controller) of our system of interest and the variety that would be required for it to maintain homeostasis. In other words, systems that want to live long must co-evolve with their environment.

More formally: we consider the environment an entity with a possible set of observable states and if two such states require different response from the system then the system must be able to differentiate between them (thus they are two different relevant states). (Note that we may not necessarily be able to describe the environment as a system, although it may contain one or more systems.)

Consequently, in Fig. 4, the complexity of a system (CS) is defined to be the complexity of the model that the controller of the system maintains (appears to be maintaining) in order to manage the system's operations, and to maintain adequate interaction with the environment.

The complexity of the system's environment (CE) is a relative notion and is defined to be the complexity of the model of the environment that the controller of the system would need to maintain the system's homeostasis (although, yet again, it is sufficient if, in the eyes of an external observer, the system's controller appears to be maintaining such model). Specifically, such an 'environment model' must have predictive capability, so that the system, while interoperating with the environment, can maintain a homeostatic trajectory in time (and space).

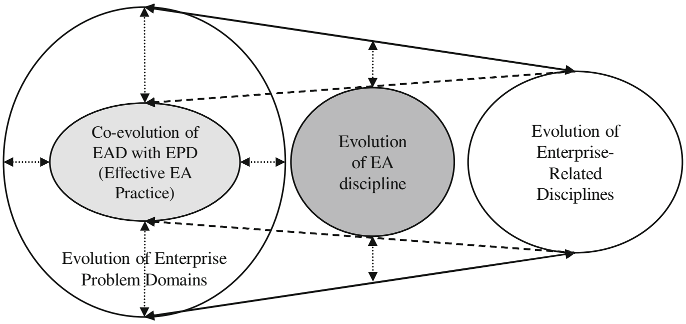


Fig. 3. An effective EA Practice: Co-evolution of EA Discipline with Enterprise Problem Domains through invocation of relevant theories, models and methods from Enterprise-Related Disciplines.

An environment model would include (a) models of external systems (including models of their controllers and operations), and (b) a model of the rest of the environment.

These models are needed to be able to represent and predict the states of signals and resources among the system, the external systems and the rest of the environment. This because based on the theorem of the 'Good Regulator' [32], a good controller of

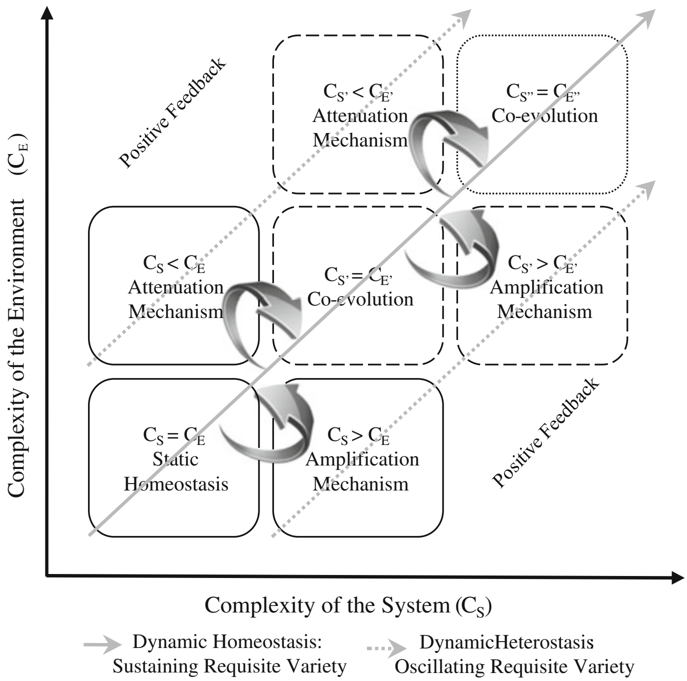


Fig. 4. Co-evolution Path Model.

a system must have a model of that system with an equal complexity at its disposal as the system to be controlled has.

Notice (Fig. 4) that (1) If the complexity of the system (CS) equals to that of its environment (CE), then the system has the requisite variety and is in static equilibrium. However, any change in the complexity of the environment should be sensed by the system's self-perpetuating mechanism to restore the system to its initial state or to create a new equilibrium state; (2) If the complexity of the environment is greater than that of the system, then the system should attenuate the effects of this complexity, i.e., change and co-evolve with its environment (in other words, the environment produced, or is recognised to have the potential to produce, some states in which the system can not function adequately); (3) If the complexity of the system is greater than that of its environment, then the system can potentially create a set of different states and perform behaviours which are not differentiated by its environment. The system can identify this extra complexity as undesired, or use an amplification mechanism to create new differentiations in the environment (e.g. marketing of new goods/services).

Enterprises as 'live systems' have a number of variables characterising essential survival properties. Ashby [25] refers to these as 'essential variables' (crucial to a system's survival) – modern literature would refer to these as critical success factors measured by strategic 'key performance indicators'. Ashby (1960) defines survival as: "… a line of behaviour [that] takes no essential variable outside given limits" [25, 33]. Therefore, by definition, any line of behaviour outside limits of essential variables is on the non-viable system path and is fatal to the system's lifeline.

For a system to be regarded as adaptive, and therefore viable, Ashby introduces two necessary feedback loops [25, 33, 34]. The first, frequently operating, feedback loop makes small parametric modifications and corrections to the system. As opposed to this, the second loop changes the structure or architecture of the system and operates if the tolerance of essential is predicted to fall outside the limits of survival. If the system's second feedback loop does not respond to the changes in complexity of the environment, then the system will be on a non-viable path.

Based on Ashby's theory of adaptation [25], Umpleby [34] indicates that the first feedback loop is necessary for a system to learn a pattern of behaviour necessary for a specific environment, while the second feedback loop is required for a system to identify the changes in the environment and design and create new patterns of behaviour.

If there is a dramatic increase in complexity of the environment on which the system is not prepared to act (due to scarcity of resources, lack of dynamic capability, inability to create new structures/adapt its architecture in a timely manner), then the lack of an appropriate second feedback loop makes the system non-viable and the system is doomed to fail.

3.2 Co-evolution Path Model of the EA Discipline

Looking at EAD as a system (the 'discipline-as-a-system') the co-evolution model of Sect. 3.1 applies to that system too, therefore the question: what are the co-evolution mechanisms through which the EA discipline can maintain its requisite variety to remain relevant in light of changes to evolving enterprise problem domains?

EA as an integrating discipline invokes models, theories, and methods of related disciplines, an effective co-evolution is only guaranteed by:
(a) invoking the right theories, models, and methods from Enterprise related Disciplines (ERD) to address new and emerging Enterprise Problem Domains (EPD) in a combined use (attenuation mechanism), and
(b) promoting new synthesised EA terminologies, reference models, and methods to provide solutions in enterprise problems domains using a holistic approach (amplification mechanism).
Thus, if at any one time the variety of the unified EA theory is less than the variety of the enterprise problem domains, then EA can not respond to the evolution of enterprise, and enterprise architecture as a discipline must increase its variety by attenuating the relevant variety through adopting new elements from relevant enterprise related disciplines.

On the other hand, an Enterprise Architect should also formulate and execute a promotion mechanism if the variety of EA models methods and frameworks is more than the variety of the enterprise problem domains. In this case, system managers, users, and stakeholders would not be able to comprehend these complex EA models, methods and etc. and would probably avoid using them in the evolution of enterprise; therefore an enterprise architect should decrease the variety of its models by amplifying the variety of the models, or promote the use of more complex models to invent solutions for the enterprise's extended 'new action domains'.

By using these mechanisms (invocation and promotion), it would be possible to sustain the co-evolution of the EA discipline and of problem domains, to ensure that EA is adaptively and effectively addressing issues of its problem domains.

The evolution of enterprise related disciplines should therefore be closely monitored so as to be able to perform the mentioned 'invocation' and 'promotion' to provide enterprise problem domains with relevant combined discipline-contributions in any EA practice (Fig. 5).

3.2.1 Example of Co-evolving/Viable Enterprise Architecture Discipline

Below are few examples presented from the history of EA, illustrating the way the needs of the environment influenced the development of EA frameworks.

The Purdue Consortium [37] was originally formed with the intention of developing a master planning methodology for designing Computer Integrated Manufacturing Systems. The development of such a methodology proved to be too hard on the basis of the Purdue CIM model [39]. It is only after the model was extended to include the human element and also extended to include both the mission fulfillment and control part of the enterprise (resulting in the Purdue Enterprise Reference Architecture (PERA) [40]) that it became possible to develop a methodology required by the environment. These two new conceptual distinctions became an important element of GERAM, in its original version dating 1994 [41, 42] (Fig. 6).

The problem arose because all methodologies sofar ignored the role of the human in the enterprise. Once the PERA framework was developed that explicitly showed the role of the human (in every pertinent life cycle phase) the variety of the AF was

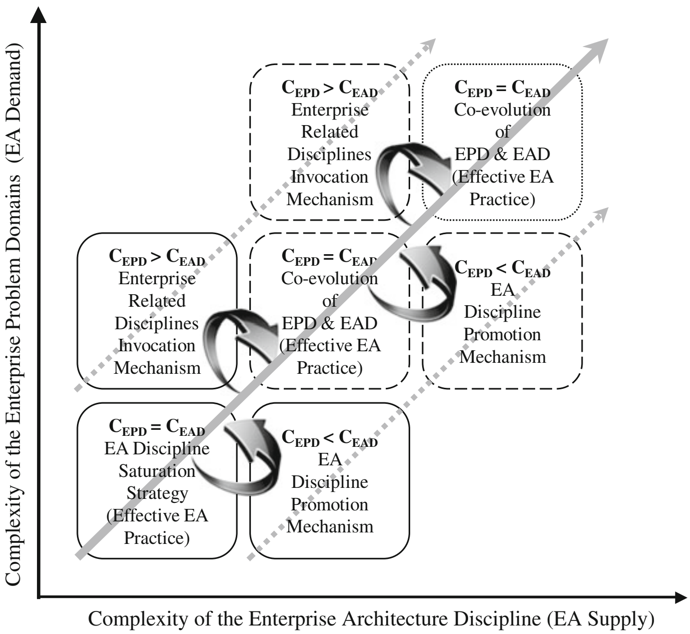


Fig. 5. Co-evolution Path Model of the Enterprise Architecture Discipline.

sufficient and the Purdue Guide for Master Planning could be written and delivered to the stakeholders.

However, in an industrial application [43] of GERAM, in the period of 1996-1998, a difficulty arose (at a Globeman 21 consortium meeting in June 1997): industry participants were under the impression that the life cycle represented the way systems were developed in time and wanted to use this view for methodology development. Due to the practical need to differentiate between temporal succession of events and the abstraction levels that design uses without imposing temporal succession, the GERAM framework had to be extended by the life history concept [44], allowing life cycle phases and life history stages to be clearly differentiated.

3.2.2 Example of Inefficient States of the Enterprise Architecture Discipline

The fact that by 1999 there existed a framework with new differentiations found necessary in some application domains (manufacturing/engineering) and that these concepts became international standard ISO15704 (under ISO TC 184 Automation systems and integration) in the year 2000 [2] did not mean that other areas (such as systems and software engineering) would have readily adopted them. From the point of view of domain specific architecture frameworks and ISO standards, they were not yet recognised as necessary in end of the 1990s and early 2000s.

In these latter environments Cs > Ce. However, the added complexity of the ISO15704/GERAM concepts *created an opportunity* in the environment.

In the manufacturing/engineering field the life cycle/life history differentiation created a new understanding of how virtual enterprises and enterprise networks relate

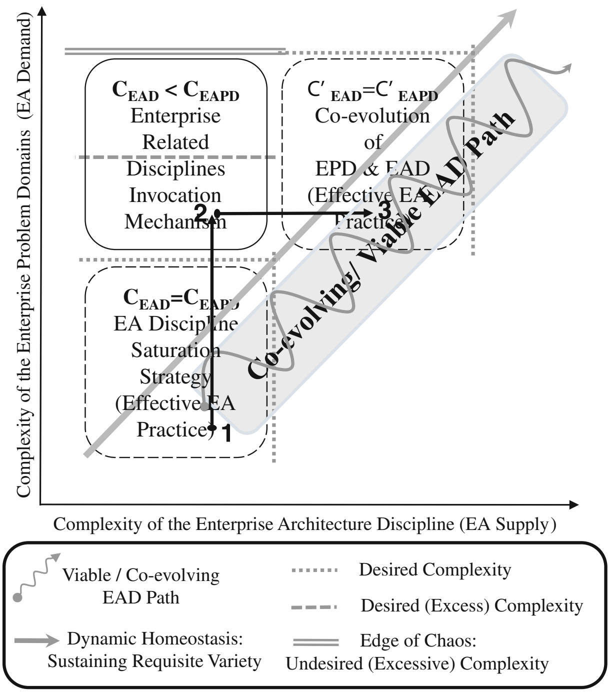


Fig. 6. Traces of Co-evolving/Viable path of EA Discipline.

and was used in real life industrial applications, such as in the GMN Consortium [44, 45]. Suddenly, by differentiating between life cycle and life history (for any entity, including enterprises, business units, programmes, projects, products) various systems development methodologies could be identified as patterns of life history. Due to the fact that each entity has its own life history, new ways of organising large system development is possible, by not having to adopt a single methodology for large scale systems engineering.

At the same time, the transfer of these differentiations from one application area (manufacturing and industrial engineering) to software and systems engineering and defense industries has not been fast. For example, it is only in the recent past that as a response to environmental needs DoDAF/MODAF EA frameworks added ‘human role as a ‘performer’ in its meta-model, and the full extent of working out how to treat the human role in several domain specific frameworks is still a work in progress.

In the case of GERAM, the introduction of the sw/hw division was anticipatory, rather than based on existing need, because at the time or the development of early EA frameworks the emphasis was on software. However, GERAM introduced the hw/sw division as an epistemological distinction orthogonal to other dimensions (such as human vs machine, and control vs mission fulfillment). This not only responded to the future threat of Ce > Cs case (even though most information systems development methodologies only looked at the software and its acceptance by the organisation) but gave a new opportunity (Cs > Ce) whereupon due to the

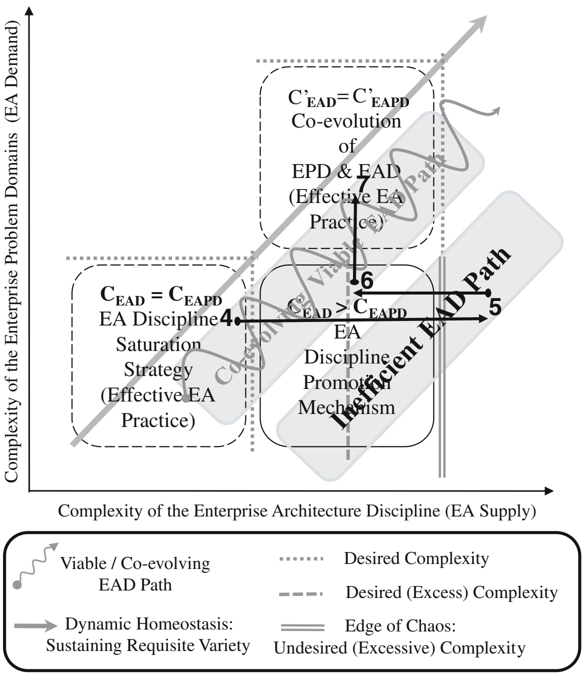


Fig. 7. Traces of Inefficient path of EA Discipline.

orthogonality of the SW/HW division to the human/machine division allowed the treatment of human sw/human hw.

The way the sw/hw aspect division was introduced allowed GERAM to represent the distinctions of knowledge management (explicit vs tacit knowledge – the 'human software' being the explicit externalized knowledge that humans can exchange and which can be transferred and stored using information technology, and the 'human hardware', the individual together with tacit as well as explicit but internalized knowledge), thus enabling communication of the knowledge management problem in a way that was consistent with the way designers of technical systems viewed the sw/hw division. The orthogonality of the sw/hw concept to other aspects was an application of an early introduction of the sw/hw mapping into EA [46] in 1985, two decades later [47] in 2006, and even though GERAM already had this distinction in [41], this orthogonality concept remained underutilised for over a decade.

3.2.3 Examples of Vulnerable Enterprise Architecture Discipline

Early EA frameworks, such as Zachman, CIMOSA, and PERA had no explicit coverage of hardware and software aspects of a system, such as an enterprise, or at least they did not have a complete coverage (what is meant by complete is explained below). Therefore, while the intention of frameworks is to provide a complete treatment of their domains, this lack for long divided the discipline – one domain covered by EA frameworks, and one by other engineering disciplines. From this state,

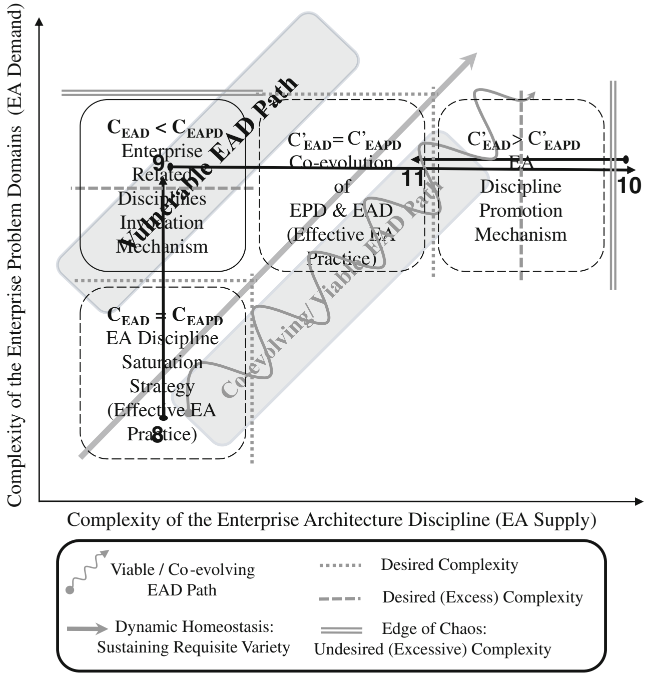


Fig. 8. Traces of Vulnerable path of EA Discipline.

there are two possible pathways: 1. the EA discipline adopting extension of its domain, due to the need of the environment, or 2. the other domain overtaking the role of EA by being extended, and making existing EA theories obsolete, and even if the response to the environment if adequate but late, scenario 2. can eventuate (Figs. 7, 8).

3.2.4 Non-viable Enterprise Architecture Discipline

In the history of science and technology there are many examples, when the theories used for understanding the world, or to develop practical methods fall so much behind the needs of the environment, that these theories become obsolete and there is no way to recover from that state (Fig. 9). Instead, completely new theories are developed. Perhaps one of the most famous examples is the series of events that led to the development of quantum mechanics, due to the inability of classical mechanics to explain (or to be extended to explain) significant real world phenomena, such as the spectrum of black body radiation. In the area of EA, a very large number of frameworks had been developed and became obsolete, as they were not extended to explain the life cycle of the entity of which the architecting was their domain. The monograph [38] includes the overview of many such architectures (called ‘architectures of type I’), with the common treat that they only explain the structure of the entity of interest, but not the life cycle of their creation. Consequently most of these ‘architectures’ are no longer in use to inform how to architect an enterprise, except for those that have been adopted as reference models.

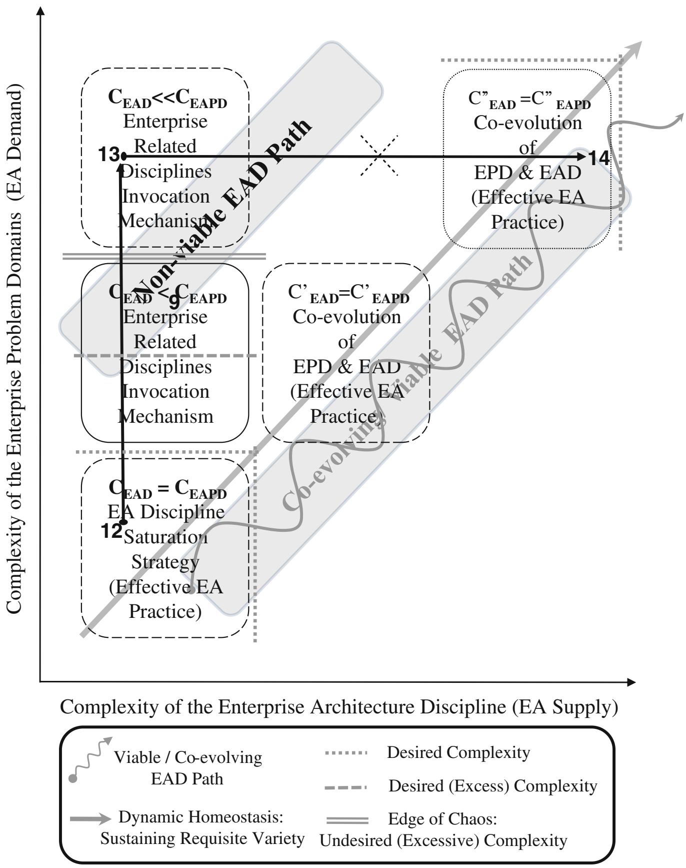


Fig. 9. Traces of Non-viable path of EA Discipline.

4 Cybernetic Model of EA as an Evolving and Developing Discipline

Enterprise Architecture, like any other developing discipline, needs a model for theory development, theory testing and knowledge creation. Anderton and Checkland [34] developed a model of any developing discipline to demonstrate the cyclic interaction between theory development and formulation for a problem, and theory testing [35, 36], (Fig. 10).

For EA to be a developing discipline (Fig. 10), we consider the real world enterprise problem domains as the source of the development process that give rise to issues that are addressed by theories, models and methods in enterprise related disciplines. These will shape ideas by which two types of theories could be developed [36]:

(a) substantive theories derived from related disciplines to apply relevant models, theories and methods in enterprise problem domains, and

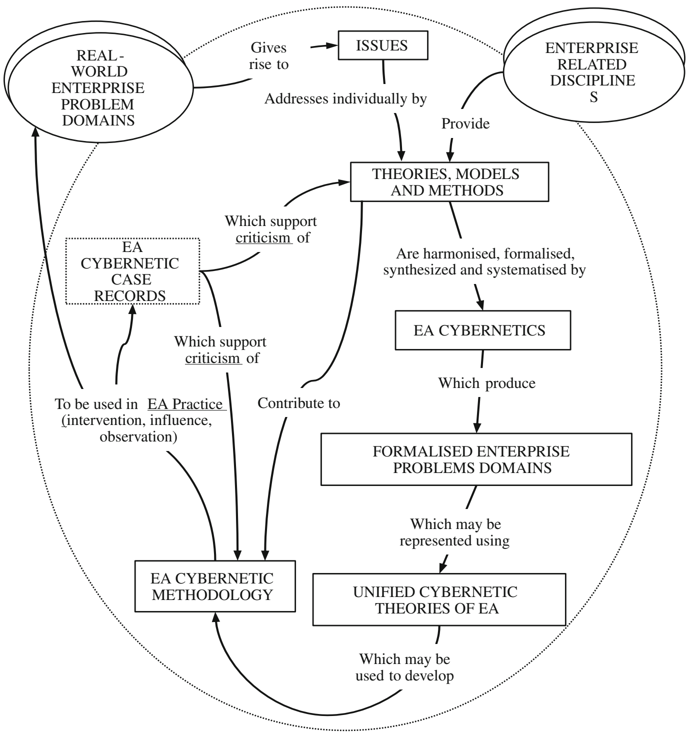


Fig. 10. A Cybernetic Model of Enterprise Architecture Discipline as a Developing Discipline based on the relationship between activities and results in a developing discipline (after [35, 36]).

(b) methodological theories about how to individually apply enterprise related disciplines in enterprise problem domains.

Once we developed such theories, we can state problems – not only existing problems in concrete problem domains, but also formalised, harmonised and synthesised problem statements by EA cybernetics within this new theory. Based on a new theory, one could express new problems and find new solutions/models never before contemplated.

A unified cybernetic theory of EA may be used to develop a corresponding methodology (or rather, methodologies) for use in EA practice. Results of such synthesis must be tested in practice (through intervention, influence, or observation) to create ‘case records’, which in turn provide the source of criticism which allow better theories to be formulated (and as a result, better models, techniques, and methodologies). The application of the latter methodologies should be documented in case records which could provide feedback to improve the individual- and the unified theories.

The EA discipline not only embraces models, methods and theories of management and control – it also uses the same from systems engineering, linguistics, cognitive science, environmental science, biology, social science and artificial intelligence.

What cybernetic thinking is able to do is to provide a method of unifying (and relating) the apport of these disciplines: cybernetic thinking can be used to represent the essence of multiple theories using abstract functions and processes (and meta-processes) and their relationships/rules/axioms (likely to be expressed in suitably selected logics).

Following the systems thinking diagram of Fig. 10, the contributions of these disciplines needs to be formalised, synthesised, harmonised, systematised and eventually represented as a unified Cybernetic Theory of EA (which we call 'EA Cybernetics').

5 Conclusions

In order to have an evolving unified theory to extend the EA Body of Knowledge with common elements that are independent from the domain and the type of change, we focused on the viability of EA as a discipline and discussed it using Beer's Viable System Model (VSM), and correspondingly introduced three basic components of a viable EA discipline using VSM.

We also proposed the concept of co-evolution mechanisms for an evolving EA discipline based on VSM and a companion theory (Ashby's law of requisite variety, but with a new, refined definition of the complexity measure for the model(s) of the environment, that takes the relativity of this term into account).

We also proposed a cybernetic model of EA as a developing discipline using Checkland's system model for a developing discipline and introduced EA Cybernetics as a distinct field of EA that harmonises, formalises, synthesises and systematises the results of systems thinking and cybernetics to enable their concerted application in EA practice.

Future work will concentrate on the application of this model of EA as an evolving discipline, whereupon testing and validation of this theoretical model is to be performed.

References

1. IFIP-IFAC-Task-Force: GERAM: generalised enterprise reference architecture and methodology, Chapter 2, Version 1(3): 6–3 (1999) In: Bernus, P., Nemes. L., Schmidt, G. (eds.) Handbook of Enterprise Architecture, Springer, Heidelberg (2003)
2. ISO15704: Industrial automation systems - Requirements for enterprise-reference architectures and methodologies. Geneva: ISO TC184.SC5.WG1. Geneva, ISO TC184.SC5.WG1 (2000, Amd.2005)
3. ISO/IEC/IEEE 42010: Systems and software engineering – Architecture description, Recommended Practice for Architectural Description of Software-intensive Systems. ISO/IEC JTC1/SC7/WG42 (2011)
4. DoDAF: DoD Architecture Framework Version 2.0, US DoD, Washington (2009)

5. TOGAF: TOGAF 1 – TOGAF 9.1 (Versions of the TOGAF Architecture Framework). Open Group (1999–2011)
6. Wiener, N.: Cybernetics or Control and Communication in the Animal and the Machine, (2nd Rev. Ed 1961). MIT Press, Cambridge (1948)
7. Ashby, W.R.: An Introduction to Cybernetics. Chapman & Hall, London (1956)
8. Beer, S.: Decision and Control: The Meaning of Operational Research and Management Cybernetics. Wiley, New York (1966)
9. Beer, S.: Diagnosing the System for Organizations. Wiley, New York (1985)
10. Boulding, K.E.: General systems theory-the skeleton of science. Manag. Sci. **2**(3), 197–208 (1956)
11. von Bertalanffy, L.: General System Theory-Foundations and Developments. George Braziller, Inc., New York (1968)
12. Pask, G.: Conversation, Cognition and Learning. Elsevier, Amsterdam (1975)
13. Doumeingts, G.: La Methode GRAI [PhD Thesis]. Bordeaux, France: University of Bordeaux I (1984)
14. Doumeingts, G.: GIM, grain integrated methodology. In: Molina, A., Kusiak, A., Sanchez, J. (eds.) Handbook of Life Cycle Engineering, Models and Methodologies, pp. 227−288. Kluwer Academic Publishers, Dordrecht (1998)
15. Suh, N.P.: The Principles of Design. Oxford University Press, New York (1990)
16. Suh, N.P.: Axiomatic Design: Advances and Applications. Oxford University Press, New York (2001)
17. Suh, N.P.: Complexity: Theory and Applications. Oxford University Press, New York (2005)
18. Holland, J.H.: Complex adaptive systems. Daedalus **121**(1), 17–30 (1992)
19. Gell-Mann, M.: Complex adaptive systems. In: Cowan, G.A., Pines, D., Meltzer, D. (eds.) Complexity: Metaphors, Models, and Reality, pp. 17−45. Addison-Wesley, Reading (1994)
20. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**(2), 115–152 (1995)
21. Wooldridge, M.J.: An Introduction to Multiagent Systems. Wiley, New York (2002)
22. Kensing, F., Simonsen, J., Bodker, K.: MUST: a method for participatory design. Hum. Comput. Interact. **13**(2), 167–198 (1998)
23. Bødker, K., Kensing, F., Simonsen, J.: Participatory IT Design: Designing for Business and Workplace Realities. MIT Press, Cambridge (2004)
24. Hammer, M., Stanton, S.A.: The Reengineering Revolution: A handbook. Harper Business, New York (1995)
25. Ashby, W.R.: Design for a Brain: The Origin of Adaptive Behavior. Wiley, New York (1960)
26. Senge, P.M.: The Fifth Discipline: The Art and Practice of the Learning Organization, Book Review. Doubleday, New York (1993)
27. Nonaka, I., Takeuchi, H.: The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, New York (1995)
28. Beer, S.: The Heart of Enterprise: The Managerial Cybernetics of Organization. Wiley, New York (1979)
29. Beer, S.: Brain of the Firm, 2nd edn. Wiley, New York (1981)
30. Kandjani, H., Bernus, P.: Engineering self-designing enterprises as complex systems using extended axiomatic design theory. In: Proceedings of the 18th IFAC World Congr. Milan, Italy, IFAC Papers On Line, vol. 18(1), pp. 11943−11948. Elsevier, Amsterdam (2011)
31. Ashby, W.R.: Adaptiveness and equilibrium. Br. J. Psychiatry **86**(362), 478–483 (1940)

32. Conant, R.C., Ashby, W.R.: Every good regulator of a system must be a model of that system. Int. J. Syst. Sci. **1**(2), 89–97 (1970)
33. Geoghegan, M.C., Pangaro, P.: Design for a self-regenerating organisation. Int. J. Gen Syst **38**(2), 155–173 (2009)
34. Umpleby, S.A.: Ross Ashby's general theory of adaptive systems. Int. J. Gen Syst **38**(2), 231–238 (2009)
35. Anderton, R.H., Checkland, P.B.: On learning our lessons, Internal Discussion Paper, Lancaster, UK, Department of Systems, University of Lancaster. 2/77 (1977)
36. Checkland, P.: Systems Thinking, Systems Practice. Wiley, Chichester (1996)
37. Industry-University Consortium, An Implementation Procedures Manual for Developing Master Plans for Computer Integrated Manufacturing. In: Williams, T.J. (ed.) Technical Report 155, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, IN, USA (1992)
38. Bernus, P., Nemes, L., Williams, T.J. (eds): Architectures for Enterprise Integration, p. 368. Chapman and Hall, London (1996)
39. CIM Reference Model Committee: A reference model for computer integrated manufacturing from the point of view of industrial automation. Int. J. Comput. Integr. Manufact. **2**(2), 114−127 (1989)
40. Williams, T.J.: The purdue enterprise reference architecture and methodology (PERA). Comput. Ind. **24**(2−3), 141–158 (1994)
41. Bernus, P., Nemes, L.: A framework to define a generic enterprise reference architecture and methodology. In: Proceedings ICARV'96 4th International Conference on Control, Automation, Robotics and Vision, vol. 3/3, pp. 88−92. Nanyang Technological University, Singapore (1994)
42. Bernus, P., Nemes, L.: A framework to define a generic enterprise reference architecture and methodology. Comput. Integr. Manuf. Syst. **9**(3), 179−191 (1996) (Elsevier, Amsterdam)
43. Zhou, M., Nemes, L., Shinonome, M., Hashimoto, H., Fuse, A., Bernus, P., Uppington, G.: A framework for design: a virtual manufacturing enterprise and its implementation. In: Mo, J. Kimura, F. (eds.) DIISM '98. IFIP TC5 WG.5.3/5.7 3rd International Working Conference on the Design of Information Infrastructure Systems for Manufacturing, pp. 339−343. Kluwer, Dordrecht (1998)
44. Vesterager, J., Bernus, P., Larsen, L.B., Pedersen, J.D., Tølle, M.: Use of GERAM as basis for a virtual enterprise framework model. In: Mo, J., Nemes, L. (eds.) Global Engineering, Manufacturing and Enterprise Networks (Proc DIISM2000), pp.75−82. Kluwer, Dordrecht (2000)
45. Tølle, M, Bernus, P.: Reference models supporting enterprise networks and virtual enterprises. Int. J. Networking Virtual Organ. **2**(1), 2−15. Inderscience, Olney Bucks (2003)
46. Létray, Z., Bernus, P.: The link between the functional and physical structure in computer integrated manufacturing. In: Proceedings APMS/Compcontrol IV. Budapest: OMIKK-Technoinform, pp. 954−964 (1985)
47. Kalpic, B., Bernus, P.: Business process modelling through the knowledge management perspective. Int. J. Knowl. Manage. **10**(3), 40–56 (2006)

A Literature Review of Business/IT Alignment Strategies

Lerina Aversano(✉), Carmine Grasso, and Maria Tortorella

Departement of Engineering, University of Sannio, Via Traiano, 82100 Benevento, Italy
{aversano, carmine.grasso, tortorella}@unisannio.it

Abstract. The alignment issue was addressed in several researches and numerous methods, techniques and tools were proposed. Choosing the most suitable approach to a specific need is a challenging task. This paper describes a characterization framework that can be applied for evaluating different alignment approaches, with the aim of discovering similarity, maturity, capability to measure, model, asses and evolve the alignment level existing among business and technological assets of an enterprise. A literature review was executed by applying the proposed framework to alignment research approach published in the Information & Management Journal and Journal of Strategic Information Systems. The achieved evaluation results are presented.

Keywords: Measurement framework · Evaluation and analysis · Enterprise evolution · Modeling · Alignment

1 Introduction

The issue of alignment was mentioned for the first time in the late 1970s and since then several studies and researches were conducted highlighting the alignment concerns - Society for Information Management (2006). Nowadays the alignment represents a top concern issue. During the last decade, several studies were proposed by researchers, practitioners and companies, but most of them are at an embryonic stage. They demonstrated through case studies, surveys and empirical approaches that the business and IT (Information Technology) performance are tightly coupled [2–5], and enterprises cannot be competitive if their business and IT strategies are not aligned. These studies regard different abstraction levels from functional to strategic [6]. In particular, Strategic Alignment of IT exists when goals, activities and processes of a business organization are in harmony with the information systems supporting them [7]. High degree of alignment positively influences IT effectiveness and leads to higher business performance [8]. In [9], the dynamic capability perspective is applied to a Taiwanese Semiconductor Company for demonstrating that it is necessary a reconfiguration of IT to support business strategy when misalignment happens. On the other hand at hand, the functional level analysis of the alignment between existing business processes and software systems is necessary for optimizing the effectiveness of the software support.

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 471–488, 2013.
DOI: 10.1007/978-3-642-40654-6_28,

Supporting and addressing future research concerning the alignment requires the knowledge of the state of the art in this area with a deep investigation of the already executed researches. With this in mind, this paper presents a systematic review of the already performed studies with reference to the alignment topic. The presented study was planned by following the comprehensive guideline that Kitchenham et al. [16]. proposed for performing a systematic literature reviews appropriate for software engineering researchers. Systematic reviews aim at presenting a fair evaluation of a research topic by using a trustworthy, rigorous, and auditable methodology. The proposed guidelines were derived from three existing guidelines used by medical researchers, for conducting a systematic review in the clinician field [17]. Performing the review required the definition of a framework for characterizing the alignment studies. The proposed characterization framework includes a set of questions aiming at understanding the goal of a proposed alignment approach and its effective applicability to a working context. The definition of the framework followed a careful analysis of the literature considering the alignment topics. This analysis aimed at identifying commonalities and differences among the proposed approaches for being later incorporated in the characterization framework. A first results of this preliminary analysis indicated that for being useful and completely applicable, an alignment strategy must include a set of components. The first step to be performed is the modeling of the various entities involved in the analysis and definition of links between business and IT entities. Then, the measurement of the alignment degree existing between the chosen assets is required for establishing if improvement actions are necessary fro improving the alignment level. Then, evolution suggestions are required for improving such a level. An automatic tool is also useful for supporting all the process of detection, assessment and evolution of the considered entities.

The framework was used for analyzing the studies published in the Information & Management Journal and Journal of Strategic Information Systems. These journals were chosen as they are the ones that published more research studies regarding the alignment concepts.

The rest of the paper is organized as follow: Section 2 describes the background of the alignment topic; Section 3 describes the proposed characterization framework; Section 4 presents the results of the analysis of the alignment papers published in the chosen journals; and final remarks are given in the last section.

2 Background

In literature, different terms are used to refer to the alignment concept. It is called: fit in [10]; bridge in [11]; integration in [12]; harmony in [13]; linkage in [14]; fusion in [15]; and further definition and terms are in [3].

A view of business and technological alignment defines at which degree the information technology mission, objectives, and plans, support and are supported by the business mission, objectives, and plans [18]. Moreover, it involves “fit” and “integration” among business strategy, IT strategy, business infrastructure, and IT infrastructure [14, 19]. A relevant “problem” is the understanding of what business and information system alignment is, how to obtain it and therefore maintain it [20].

Traditional approaches addresses the alignment concern for understanding how organizations can achieve alignment, but little contribution is given regarding how to identify and correct misalignment.

Understanding an organization's alignment level requires the modeling of the involved entities. Different models are introduced in literature. One of them was SAM – Strategic Alignment, Model from Henderson and Venkatraman (1993). Different study were later performed for evaluating these models. For example, in [21] the SAM model was used in financial service firms for determining if it was useful to asses strategic alignment between IT and business. In [22], the general aspects concerning modeling was well debated and a modeling issue was proposed. In particular, the VMOST – Vision, Mission, Objectives, Strategies, Tactics – analysis was treated to split the business strategy into the main components of vision, mission, goals, strategies and tactics, and the BRG – Business Rules Group – model was proposed for modeling the organization's systems. In [23], the MDA – Model Driven Architecture – tool was used to support the alignment management, and meta-models were proposed for representing the entities involved in the alignment analysis. In [24], a framework was proposed for modeling the alignment at the functional level and some metrics were introduced for measuring the alignment degree between business processes and software systems. In [25], criteria and associated generic metrics were proposed to quantify at which extent there is a fit between the business and system which supports it. In [26], a framework was presented for analyzing the alignment problem and proposing an approach to application architecture design with reference to a business context.

The Business and Information Systems MisAlignment Model (BISMAM), was proposed in [18, 27], to understand, classify and manage misalignments. The proposal addresses the alignment problem combining the misalignment approach with medical sciences approaches, based on a metaphor between misalignment and disease. The authors believe that the misalignment approach is closer to organizations real life and that medical sciences approaches provide relevant concepts and techniques for misalignment classification and management.

The research constructs were measured using multi-item scales adapted from the SAM framework [28]. The relationship existing between the alignment maturity dimensions and IS strategic alignment was examined and the results were applied to provide a snapshot of business–IT alignment in China. In [29], a new conceptualization of alignment was reported together with the development and testing of a parsimonious model which addresses this issue. Data from a survey of 415 respondents from medium-large New Zealand companies were used to test the model. It was found that IS-marketing alignment had a positive impact on both business and marketing performances, and that the latter had a modest but positive impact on business performance. This study extended the application of Venkatraman's from [6], and offered a support to the robustness of his conceptualization and measurement of strategic orientation.

3 The Characterization Framework

The alignment strategies proposed in literature consider different aspects of the alignment and analyze it at different abstraction levels. The aim of the proposed framework is to understand if an alignment strategy is suitable to an enterprise's specific needs and, in particular, if its description is complete and clear for easily applying it. The proposed framework was defined for being generally applicable for analyzing any kind of alignment strategy. Then, the main components it considers represent a synthesis of all the aspects covered by the alignment strategies proposed in literature. Specifically, the framework considers the three following main phases [24]:

1. **Modeling.** All the entities involved by the alignment analysis should be modeled, so to exclude all the business and technological details that are not relevant for the study. This phase is necessary to search and represent the information that the considered alignment approach uses for analyzing the alignment at the considered abstraction level. The modeled entities regarding the different aspects involved in the alignment evaluation should be mapped, so to facilitate the next analysis.
2. **Alignment Evaluation.** An alignment approach should quantitatively evaluate the alignment degree of the considered entities for objectively analyzing it and understanding if it reaches a satisfying level or improvement actions should be performed for increasing it. This requires the use of suitable and easily quantifiable metrics.
3. **Evolution Execution.** If the alignment level does not reach a satisfying level, a misalignment exists in the analyzed entities, and evolution actions should be performed and for increasing it.

The proposed characterization framework considers each of the activities cited above by including, for each of them, a section with a set of questions. In addition, the framework includes an initial section of generic questions aiming at categorizing the alignment approach discussed in a considered research paper and capturing the generic information regarding it.

Every question is formulated so that it can be answered by analyzing the documentation of a considered strategy and using the following values:

- Yes, indicating that the information required by the question is clearly and completely described in the analyzed documentation.
- No, indicating that the analyzed documentation does not consider the specific aspect the question concerns.
- Partially, indicating that the aspect indicated in the question is only partially addressed in the documentation.
- Not clear, indicating that the documentation does not clearly describe the information needed for answering the question.
- Not defined, indicating that the documentation does not describe cite the information needed for answering the question.

The following sub-sections describe the four sections of the characterization framework detailing the questions introduced in each of them.

3.1 General Questions

The first top-level questions deal with general aspects of an alignment strategy and aims at categorizing it. Table 1 lists the questions included in this section of the framework. In particular, the questions are formulated for understanding if motivations, needs of the alignment analysis (D1), and challenges of the considered strategy (D4) are clearly debated. Alignment strategies can analyze this aspect at different levels, involving diverse entities. In fact, regarding the business assets, the strategy can consider: enterprise goals, business entities, business strategies and business processes; on the other side, from the Information Technology point of view, it is possible to consider technologies and information systems (applications and data) [31]. These entities are considered at two different: *strategic level* analyzing business strategy and IT strategy, and *functional level* considering business processes and information systems [14]. Then, the proposed framework also includes questions for understanding these aspects (D2, D3, D5). In addition, the definition of pre-conditions for applying the strategy are investigated, as some information can be missing in the operative context for being able to apply the analyzed approach (D6). Furthermore, the initial section of the framework analyses if: the strategy was defined by considering previous experiences and underwent to improvement actions (D7); it included quantitative studies (D8), and it suggested improvements and extensions in the future (D9). All this helps to understand its scientific maturity; while the experimental maturity is verified by considering the application on the field of the considered approach and knowledge and experience gained through its use (D10).

3.2 Modeling Questions

The second section of the framework includes questions dealing with modeling activities. Table 2 reports the included questions aiming at investigating the completeness of the available information regarding the existence of modeling techniques in the alignment approach described in an analysed research study (D11), and the possibility of modeling the elementary entities involved in the alignment analysis and

Table 1. General questions.

Id	General
D1	Are motivations and needs of the alignment strategy treated?
D2	Is the dimension of the considered type of alignment discussed?
D3	Is the concept of alignment defined?
D4	Is the challenges in attaining the treated type of alignment?
D5	Are the entities involved in the considered type of alignment discussed ?
D6	Are the pre-conditions for applying the proposed strategy clearly stated?
D7	Is the proposed strategy scientifically mature?
D8	Is the need of quantitative methods discussed?
D9	Are future perspectives and/or future work proposed?
D10	Are lessons learned discussed?

Table 2. Modeling questions.

Id	Modeling
D11	Are models to represent alignment used?
D12	Are models to represent the separate entities used?
D13	Is the proposed model based on existing research approaches?
D14	Is the modeling automatically supported?
D15	Was the proposed modeling approach applied to case studies?

related reciprocal relationships (D12). Moreover, the questions investigate on the maturity of the analyzed modeling approach by verifying if its definition depends on other approaches (D13) and it was already applied to case studies (D15) or working contexts (D16).

3.3 Alignment Evaluation Questions

The third group of questions concerns the alignment measurement activity. Table 3 presents the questions formulated with the aim of verifying if the approach described in the analyzed paper includes measurement activities and the related description (D17). As the adoption of already existing techniques may contribute to increase the effectiveness of an approach, question D18 considers this aspect; while the subsequent questions (D19, D20) ask if the approach was previously applied for understanding its applicability. Finally, specific questions are defined for understanding the exploitation of statistical methods (D21) and automation level of the proposed approach (D22).

3.4 Alignment Evolution Questions

The last questions of the framework are presented in Table 4 and regards the possibility that an analyzed paper considers evolution activities for managing the alignment of the analyzed entities and to be performed when misalignment happens (D23). Even this group of questions analyzed the maturity and applicability of the proposed approach. In particular, questions regarding the definition of the proposed evolution support is based on the current research literature (D24), that it is automatically

Table 3. Alignment evaluation questions.

Id	Measurement
D17	Is a method to measure the level of alignment utilized?
D18	Is the proposed alignment measurement method based on existing research approaches?
D19	Is the alignment measurement method applied to case studies?
D20	Was the alignment measurement method applied on the field?
D21	Are statistical analysis used and the results summarized?

Table 4. Alignment evolution questions.

Id	Evolution
D23	Is an approach proposed for addressing and evolving the alignment?
D24	Is the proposed evolution approach based on existing research approaches?
D25	Is the evolution approach automatically supported?
D26	Was the evolution tool applied to a case studies?
D27	Was the evolution tool applied on the field?

supported (D25), and it was already applied in operative contexts (D26 and D27), are considered.

4 Applying the Framework

The proposed characterization framework was applied for analyzing some alignment approaches selected from the literature. In particular, a full investigation of the research papers concerning alignment was performed. Numerous journal and conference papers were identified. Therefore, it was decided to concentrate the attention on journal papers as they should publish more mature research results. With this in mind, the IEEE, ACM, Springer, Elsevier and Science Direct database were queried. The journal most representative of the alignment topic were the **Information & Management Journal and Journal of Strategic Information Systems from Elsevier**. 28 articles were identified in the first Journal: 5 of them were not available online, while 4 papers were discarded as they did not concern the Business and IT alignment and 2 of them were published before 2000. **23** articles were identified in the Journal of Strategic Information Systems: 8 of them were discarded because published before 2000, and 4 because they were out of topic. In total, 28 papers listed in the Appendix were analyzed. The paper codes indicated in the appendix are used in the following for referencing the papers.

The alignment topic was treated since 1995, but just papers published since 2000 are considered in this paper. In particular, those papers focused on strategic alignment during Business and Information System planning. Some of them are industrial survey. For example papers S18, S27 and S28 focalizes their attention on the strategic alignment, and collect data and use statistical analysis for evaluating and classifying the degree of alignment. After publishing these initial papers, the alignment topic was not considered since 2000, and only after 2003, it was possible to observe a growing interest regarding these aspects. Table 5 shows a classification of the papers on the

Table 5. Classification of the papers.

Type	Paper
Practice	S2, S5, S6, S7, S8, S9, S11, S12, S13, S16, S17, S1, S20, S26, S28, S18,S19
Research	S3, S5, S10, S11, S14, S15, S17, S4, S21, S22, S23, S24, S27
Review	
Survey	

basis of their kind. Four categories were considered: Practice, Research, Review and Survey. The table shows that the papers regard practice and research and no review or survey was found. In particular, the large part of the analysed papers are practical, even if some of them, such as S5 and S17, face the alignment problem from the point of view of the research by proposing new approaches.

Before analysing all the obtained evaluation results, the application of the characterization framework is shown in Table 6 reporting the answers collected by applying the characterization framework to paper S11 and the explanation to each answer. The analysis of the answers highlights that the paper concerns strategic alignment. The paper appears to miss many aspects. In particular, methodologies for supporting alignment modeling, evaluation and evolution are not provided. The paper rather appears to be an empirical study regarding IT-enabled flexibility, competitive impacts, and organizational moderators of business value.

Tables 7 and 8 include the answers to all the questions coming of the characterization framework after its application to the analysed papers. The tables uses: *part* for the "partially" answer; *ndef* for the "not defined" answers; and *ncl* for the "not clear" answers. For the sake of clarity, the "yes" and "part" answers are shadowed. Tables 7 and 8 show that no paper describes an approach including all the three activities considered in Sect. 3. All the papers provide general information regarding the proposed approach even if they very often do not include sufficient details for understanding its usefulness, maturity, advantages and future perspective.

Table 9 includes the distribution of the analysed papers with reference to the considered dimension. The large part of the papers considers the alignment at the strategic level and only some at the functional level. Some approaches, such as S3, S15 and S17, consider both levels, strategic and functional. All the papers, but S17, describe the entities involved in the approach they propose, as shown in Table 10. It can be noticed that the large part of the proposed approaches considers business entities, with particular reference to business strategies and processes; while, few of the analysed approaches also consider the IT components.

Tables 7 and 8 show that few papers include evaluation activities, showing that few attention is paid to the measurement activities. The approaches considering this aspect often base their solution on existing approaches. In particular, Table 11 includes the main adopted measurement approach. STROEPIS – Strategic Orientation of the Existing Portfolio of IS applications – is a measurement model based on the eight dimensions STROBE – Strategic Orientation of Business Enterprises – instrument. In [3], the authors characterized the strategic alignment as the fit between STROBE and STROEPIS. The Balanced scorecard (BSC), used in S9, translates an organization's mission and strategy into a comprehensive set of performance measures that provide the framework for a strategic measurement and management system. It measures organizational performance across four balanced perspectives: financial, customers, internal business processes, and learning and growth. The BSC enables companies to track financial results, while simultaneously monitoring progress in building the capabilities and acquiring the intangible assets they need for future growth. On the other side, many papers present empirical studies and Table 12 lists the approaches used for performing the statistical analysis in the considered papers. Actually, even Table 14 indicated that many papers pay attention to the execution of

Table 6. Results from the analysis of papers S11.

	General questions
D1	**yes:** Business, public, and governmental organizations confronted with time and other pressures must adjust their strategies, but change cannot be accomplished unless the IT Infrastructure (ITI) is accommodated in an efficient and effective manner.
D2	**yes:** Strategic Alignment
D3	**No**
D4	**yes:** As ITI investments are not always guided by current business needs, efforts to extend ITI should consider how flexibility is introduced into each of its elements and how they are interrelated. The approach therefore identifies the sources of flexibility and their interrelationships and find how they are related to the perceived IT value.
D5	**yes:** The entities involve are: technical ITI elements, human ITI elements, process ITI element
D6	**not defined**
D7	**No**
D8	**No**
D9	**No**
D10	**yes:** The identified lesson learned brought to the identification of three limitations of the approach: such a research design only establishes associations between constructs, whereas causality must rely on theoretical justification; although organizational IT users may find the evaluation of ITI resources and capabilities difficult, their perspective is necessary to identify gaps in different perceptions of ITI; the dynamics of longitudinal processes cannot be analysed using this methodology.
	Modelling questions
D11	**yes:** The approach just hypothesized the use of research model
D12	**No**
D13	**not clear**
D14	**No**
D15	**yes:** Empirical study
D16	**No**
	Measurement questions
D17	**No**
D18	**No**
D19	**yes:** Web-based survey
D20	**No**
D21	**yes:** SEM - Structural Equations Models- techniques and MLE - Maximum Likelihood Estimation (MLE), one-way ANOVA, CFI, ComparativeFit Index, RMR, Root Mean square Residual, RMSEA, Root Mean Square Error of Approximation, Chi-square, AGFI adjusted GFI(Goodness-of-fit)
D22	**No**
	Evolution questions
D23	**No**
D24	**No**
D25	**No**
D26	**No**
D27	**No**

empirical activities. They are very often conducted for analysing some trends and situations in a set of analysed organizations. Among the used statistical approach cited in Table 12, SEM – Structural Equation Modeling – is a statistical technique allowing the researcher to test hypothesized direct relationships between independent and dependent variables, such as multiple regression, and allowing the testing of indirect or mediated relationships between observed and unobserved latent variables, while

Table 7. Results of the analysis of the considered papers from Journal Information and Management.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17
D1	yes	yes	yes	part	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
D2	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
D3	no	no	yes	no	yes	yes	part	part	part	yes	no	no	part	no	yes	no	part
D4	yes	no	yes	no	yes	yes	yes	yes	yes	yes	yes	part	part	part	yes	yes	yes
D5	yes	yes	yes	yes	part	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
D6	ncl	yes	par	yes	yes	yes	no	yes	yes	yes	ndef	part	no	part	ndef	no	no
D7	no	no	ncl	yes	yes	yes	no	yes	part	yes	no	no	yes	no	ndef	yes	yes
D8	ndef	no	yes	no	yes	yes	no	part	yes	yes	no	part	yes	no	ndef	no	part
D9	no	part	no	part	yes	yes	no	no	no	no	no	part	no	yes	ndef	no	yes
D10	yes	no	ncl	no	no	yes	no	yes	no	yes	yes	no	yes	yes	yes	yes	no
D11	yes	yes	yes	yes	yes	yes	no	no	part	yes	yes	no	yes	yes	yes	yes	no
D12	no	no	yes	no	no	yes	no	no	yes	yes	no	no	yes	no	no	yes	no
D13	no	yes	ncl	no	no	yes	yes	no	yes	ncl	ncl	no	yes	yes	yes	ncl	no
D14	no	yes	no	part	no	no	part	no	no	no	no	no	no	no	no	no	no
D15	no	yes	yes	yes	no	yes	no	no	no	no	no	no	no	yes	no	no	no
D16	yes	no	yes	no	yes	yes	yes	no	yes	yes	yes	no	yes	no	no	yes	yes
D17	yes	no	yes	no	yes	no	yes	yes	yes	yes	no	no	no	no	no	no	no
D18	ncl	no	ncl	no	yes	no	no	yes	yes	yes	no	no	no	no	no	no	no
D19	no	no	yes	no	no	no	no	no	no	no	no	no	no	no	no	no	no
D20	yes	no	yes	no	yes	no	yes	yes	yes	yes	yes	yes	yes	no	no	yes	yes
D21	yes	yes	part	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	yes	no
D22	no	no	yes	no	yes	no	yes	no	no	yes	no	no	no	no	no	no	no
D23	no	no	yes	no	part	yes	yes	yes	part	yes	no	no	no	no	no	no	no
D24	no	no	part	no	yes	yes	yes	yes	yes	ncl	no	no	no	no	no	no	no
D25	no	no	part	no	part	no	yes	no	no	no	no	no	no	no	no	no	no
D26	no	no	yes	No	no	no	no	no	no	no	no	no	no	no	no	no	no
D27	no	no	yes	No	yes	yes	no	yes	yes	yes	no	no	no	no	no	no	no

examining the reliability of the items to the latent variables. LISREL – Linear Structural RELations – is the most general program that is available for estimating structural equation models.

It can be used to analyze data from survey, experiments, experimental designs, and longitudinal studies. It allows one to test the goodness of fit of models, diagnose problems with models, fix or constrain model coefficient, do multiple-group analyses, estimate means and intercepts as well as slopes, and most importantly, distinguish between latent concepts and observed indicators.

The analysed papers also give importance to the modelling activities. Many of them are based on already existing modelling approaches.

Table 8. Results of the analysis of the considered papers from Journal of Strategic Information Systems.

	S18	S19	S20	S21	S22	S23	S24	S25	S26	S27	S28
D1	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
D2	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
D3	no	no	yes	no	no	no	no	no	yes	part	no
D4	yes	yes	yes	yes	yes	yes	yes	part	no	yes	yes
D5	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
D6	no	yes	yes	yes	yes	part	ndef	part	no	yes	no
D7	no	no	yes	no	no	part	part	yes	ndef	no	yes
D8	no	yes	no	no	no	part	no	yes	no	no	yes
D9	yes	part	part	part	yes	yes	yes	yes	yes	yes	yes
D10	yes	yes	part	yes	yes	yes	yes	yes	yes	yes	yes
D11	no	no	yes	yes	yes	yes	yes	yes	no	no	no
D12	no	no	yes	yes	no	yes	yes	yes	no	no	no
D13	no	no	yes	yes	yes	part	part	yes	no	no	no
D14	no	no	no	no	no	no	no	no	no	no	no
D15	no	no	yes	yes	yes	no	no	no	no	no	no
D16	no	no	no	no	no	yes	yes	yes	no	no	no
D17	no	yes	no	no	no	no	no	yes	no	no	yes
D18	no	yes	no	no	no	no	no	yes	no	no	yes
D19	no	yes	no	no	no	no	no	no	no	no	no
D20	no	no	no	no	no	no	no	yes	no	no	yes
D21	yes	no	no	no	no	no	no	no	no	yes	yes
D22	no	no	no	no	no	no	no	no	no	no	no
D23	no	no	no	no	no	no	no	no	no	no	no
D24	no	no	no	no	no	no	no	no	no	no	no
D25	no	no	no	no	no	no	no	no	no	no	no
D26	no	no	no	no	no	no	no	no	no	no	no
D27	no	no	no	no	no	no	no	no	no	no	no

Table 13 describes the used techniques with reference to the paper using them. Many papers considers the SAM – Strategic Alignment Model – model [14]. It is useful to treat the IS strategy alignment and support a collaborative process between the business strategy, business organisation, IS infrastructure, and IT strategy, at two different abstraction level of the alignment: functional and strategic. The Path model is used to organize different variables. In particular, in S9, hypotheses are considered, having as a starting point the importance of the strategic alignment, and motivations and success of the ERP projects. The model captures the relationships between the degree of success of ERP projects, the associated business process changes, and subsequent internal efficiency benefits.

Then, it captures the impact of internal process efficiency on customer and financial benefits. Paper S10 adopts the gestalt research model considering a

Table 9. Distribution of the papers with reference to dimension.

Dimension of alignment	Paper
Strategic	S2, S3, S5, S6, S7, S8, S9, S11, S12, S13, S15, S17, S1, S20, S21, S23,S24, S26, S27, S28, S18
Functional	S3, S14, S15, S16, S17, S4, S20, S25

Table 10. Involved entities.

Involved Entities	Paper
Business strategy	S2, S5, S8, S7, S9, S10, S25, S13, S1, S20, S26, S23, S21, S24, S27, S18, S19
IT strategy	S2, S5, S8, S10, S13, S20, S25, S26, S23, S21, S27, S28, S18, S19
IT investment	S8, S13
Business performance	S7, S8, S1
Business structure	S10, S23, S21
IT Structure	S10, S13, S23, S21
Business process	S3, S13, S15, S22
Organization's structure	S13, S20
Human resource	S15
ERP strategy, time cost, financial benefits	S9, S22
Critical success factor	S3
IT systems	S3, S5
IM (Information Management) Strategy	S28
Business objectives, E-business performance, E-commerce strategy, E-commerce strength and opportunities	S1
Business rule	S14, S15
Service systems	S14
Environmental uncertainty, information intensity, business dependence on It, IT participation in business planning, IT plan, business plan, competitive advantage	S16
IS managers, systems development methodologies	S17
Goal (enterprise level), functional (scenario level), data, output misfits (activity level)	S4
IS strategy, corporate strategy	S24, S28
Organization's IS	S7
Technical elements of IT infrastructure, Human elements of IT infrastructure, Process elements of IT infrastructure	S11, S20
IS/IT manager, business manager	S2
Infrastructure, application	S5
Internal IT subunit alignment	S25

Table 11. Used measurement approaches.

Measure	Paper
BSC, balanced score card	S9
STROBE, strategic orientation of business enterprises	S7, S8, S9, S28
STROEPIS, strategic orientation of the existing portfolio of IS applications	S7, S8, S9, S28
STROIM, strategic orientation of information management	S28
Other	S5, S25, S19

Table 12. Considered Statistical Analysis Approaches.

Statistical Analysis	Paper
ANOVA, One-way analysis of variance	S10, S11, S1
CFI, Comparative Fit Index	S6, S7, S11, S13, S16, S18
NNFI, Non-Normed Fit Index	S7, S13, S16, S18
Satorra–Bentler (SB x2/d.f.)	S13
RMR, Root Mean square Residual	S6, S7, S11, S13, S16, S28, S18
RMSEA, Root Mean Square Error of Approximation	S6, S7, S10, S11, S13, S16
Correlations	S10
Error variances	S10
GIF, Goodness-of-Fit Statistic	S6, S28
ROA, average Return-On-Assets	S12, S1
ROS, PNP	S1
TLI, Tucker–Lewis Index	S16
Chi square	S6, S10, S11, S16, S28, S18
Chi square/d.f.	S6, S7, S16, S18
AGFI adjusted GFI (Goodness-of-fit)	S7, S11, S28
LISREL (linear structural relations)	S7
SEM technique is a statistical structural equation modeling	S11
Two-tailed F-test	S2
Standard deviation	S27
Other	S5, S17

perspective of fit, and looking at a large number of variables that collectively define a meaningful and coherent slice of organizational reality. The Business rules services model is considered in S14. It provides high level services and functions that evolve during the maturity and expanded the scope of the business rules deployments across an enterprise. The Business Rules Deployment Maturity Model identifies maturity along five dimensions, including organizational scope, ownership, structure, development responsibility, and implementation responsibility. In addition, many analysed papers define their own measurement approach.

Table 7 shows that few papers (just S3, S5, S6, S8, S9 and S10) deals with the evolution the considered entities. This demonstrated that the attention is nowadays more concentrated in knowing what alignment is and how to manage it.

Table 13. Used modelling techniques.

Model	Paper
SAM strategic alignment model	S2, S6, S15, S20, S21, S23
Path model	S9
Gestalt model of strategic alignment	S10
Business rules deployment maturity model	S14, S16
Business rules tasks/services model	S14
UML model	S4
VRIO framework, the resource-based model of sustained competitive advantage	S22
Integrated model of alignment within the IT unit (adapted from Luftman and Kempaiah (2007))	S25
(CBM), Component Business Model, (USM), Unified Service Model	S27
Other	S3, S7, S11, S13, S1, S24

Table 14. Application of the proposed approach.

Type	Paper
Case study	S2, S3, S14, S4, S21, S23, S19
On the field	S3, S5, S6, S7, S10, S11, S12, S16, S1, S13, S25, S28, S27
Empirical study	S5, S6, S7, S8, S9, S10, S11, S12, S13, S16, S17, S1, S20
Example	S22, S24

Finally, many papers apply the proposed approach as indicated in Table 14. The main attention is paid to applications on the field and empirical studies.

5 Conclusions

The alignment between business and information systems assumed a growing relevance in the last years. This research issue was addressed in several researches proposing numerous methods, techniques and tools. This paper proposes a characterization framework to characterize different approaches, with aim of discovering similarity, maturity, capability to measure, model, asses and evolve the alignment. This kind of investigation is aimed to support and address future research concerning the alignment. Indeed, it is necessary to understand which are the aspects considered in the literature of this area with a quantitative approach. Because the field of alignment is wide and concerns different aspects, the aim of the presented study is to help practitioners, students and researchers to focalize the attention on a particular interested issue.

The proposed characterization framework was applied to the research works regarding the alignment topics published in Journal Information & Management and Journal of Strategic Information Systems, and the results of the evaluation is presented. The results for this preliminary application of the characterization framework emphasize that the modeling, measurement and evolution phases of an alignment approach are not adequately addressed in the analyzed strategies. Obviously, besides the Journal Information & Management, many other sources of alignment approaches exist and the results obtained in this preliminary study need the confirmation of a wider investigation involving more and more research approaches. This will be one of the main future work on which the authors are working.

As further future work, the framework proposed can be used to make a survey of the studies presented in the literature, and understand how to better address the research issues in the alignment area. The aim will also regard the classification of different model, measurement, and quantitative approaches addressing the alignment issue at different abstraction level, and understanding which of them better address a specific alignment problem.

A.1 Appendix: The Analysed Papers

Considered papers from Journal Information & Management – Elsevier

S1. Kearns, G. S.: An electronic commerce strategic typology: insights from case studies. 42(7), pp. 1023–1036, October 2005.

S2. Burn, J. M., Szeto, C.: A comparison of the views of business and IT management on success factors for strategic alignment. 37(4), pp. 197–216, June 2000.

S3. Peak, D., Guynes, C.S., Kroon, V.: Information technology Alignment Planning—a case study. 42(4), pp. 619–633, May 2005.

S4. Wu, J., Shin, S., Heng, M. S. H.: A methodology for ERP misfit analysis . 44(8), pp. 666–680, December 2007.

S5. Cumps, B., Martens, D., De Backer, M., Haesen, R., Viaene, S., Dedene, G., Baesens, B., Snoeck, M.: Inferring comprehensible business/ICT alignment rules. 46(2), pp. 116–124, March 2009.

S6. Chen, L.: Business–IT alignment maturity of companies in China. 47(1), pp. 9–16, January 2010.

S7. Johnson, A. M., Lederer, A. L.: CEO/CIO mutual understanding, strategic alignment, and the contribution of IS to the organization. 47(3), pp. 138–149, April 2010.

S8. Byrd, T. A., Lewis, B. R., Bryan, R. W.: The leveraging influence of strategic alignment on IT investment: An empirical examination. 43(3), pp. 308–321, April 2006.

S9. Velcu, O.: Strategic alignment of ERP implementation stages: An empirical investigation. 47(3), pp. 158–166, April 2010.

S10. Bergeron, F., Raymond, L., Rivard, S.: Ideal patterns of strategic alignment and business performance. 41(8), pp. 1003–1020, November 2004

S11. Fink, L., Neumann, S.: Exploring the perceived business value of the flexibility enabled by information technology infrastructure. 46(2), pp. 90–99, March 2009.

S12. Choe, J.: The effect of environmental uncertainty and strategic applications of IS on a firm's performance, 40(4), pp. 257–268, March 2003.

S13. Newkirk, H. E., Lederer, A. L.: The effectiveness of strategic information systems planning under environmental uncertainty. 43(4), pp. 481–501, June 2006.

S14. Nelson, M. L., Peterson, J., Rariden, R. L., Sen, R.: Transitioning to a business rule management service model: Case studies from the property and casualty insurance industry. 47(1), pp. 30–41, January 2010.

S15. Aerts, A. T. M., Goossenaerts, J. B. M.: Hammer, D. K., Wortmann, J. C.: Architectures in context: on the evolution of business, application software, and ICT platform architectures. 41(6), pp. 781–794, July 2004.

S16. Kearns, G. S., Lederer, A. L.: The impact of industry contextual factors on IT focus and the use of IT for competitive advantage. 41(7), pp. 899–919, September 2004.

S17. Huisman, M., Iivari, J.: Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. 43(1), pp. 29–49, January 2006.

Considered papers from Journal of Strategic Information Systems – Elsevier

S18. Kearns, G.S., Lederer A.L., 2000. The effect of Strategic alignment on the use of IS-based resources for competitive advantage, 9 pp. 265–293.

S19. Kim, S.H., Jang, D.H., Lee, D.H., Cho, S.H., 2000. Methodology of constructing a decision pathfor IT investment, 9 pp. 17–38.

S20. Avison, D., Jones, J., Powell, P., Wilson, D., 2004. Using and validating the strategic alignment model, 13 pp. 223–246.

S21. Wijnhoven, F., Spil, T., Stegwee, R., Tjang A Fa. R.,2006. Post-merger IT integration strategies: An IT alignment perspective, 15 pp. 5–28.

S22. Beard, J., W., Sumner, M., 2004. Seeking strategic advantage in the post-net era: viewing ERP systems from the resource-based perspective, 13 pp. 129–150.

S23. Boonstra, A., Broekhuis, M., van Offenbeek, M., Wortmann, H.,2011. Strategic alternatives in telecare design Developing a valueconfiguration-based alignment framework, 20 pp. 198–214.

S24. Peppard, J., Ward, J., 2004.Beyond strategic information systems: towards an IS capability,13 pp. 167–194.

S25. Dhaliwal, J., Onita, C., G., Poston, R., Zhang, X., P., 2011. Alignment within the software development unit: Assessing structural and relational dimensions between developers and testers, 20 pp. 323–342.

S26. Mohdzaher B. Mohdzain, John M. Ward, 2007. A study of subsidiaries' views of information systems strategic planning in multinational organizations, 16 pp. 324–352.

S27. Cragg, P., King, M., Hussin H., 2002. IT alignment and firm performance in small manufacturing firms, 11 pp. 109–132.

S28. Ragu-Nathan, B., Ragu-Nathan, T.S., Tu, Q., Shi, Z., 2001. Information management (IM) strategy: the construct and its measurement 10 pp. 265–289.

References

1. Society for Information Management: IT Management Concerns Survey. What keeps CIO awake at night? (2006)
2. Chan, Y.E., Huff, S.L., Barclay, D.W., Copeland, D.G.: Business strategic orientation, information systems strategic orientation, and strategic alignment. Inf. Syst. Res. **8**(2), 125–150 (1997)
3. Chan, Y.E., Reich, B.H.: State of the art IT alignment: what have we learned? J. Inf. Technol. **22**, 297–315 (2007)
4. Kearns, G.S., Lederer, A.L.: A resource-based view of strategic IT alignment: how knowledge sharing creates competitive advantage. Decis. Sci. J. **34**(1), 1–29 (2003)
5. de Leede, J., Looise, J.C., Alders, B.: Innovation, improvement and operations: an exploration of the management of alignment. Int. J. Technol. Manag. **23**(4), 353–368 (2002)
6. Henderson, J.C., Venkatraman, N.: Strategic alignment: leveraging information technology for transforming organizations. IBM Syst. J. **32**(1), 4–16 (1993)
7. McKenn, J.D., Smith, H.: Making IT Happen: Critical Issues in IT management. Wiley, Chichester (1993)
8. Croteau, A.M., Bergeron, F.: An information technology trilogy: business strategy, technological deployment and organizational performance. J. Strateg. Inf. Syst. **10**, 77–99 (2001)
9. Chen, R., Sun, C., Helms, M., Jih, W.: Aligning information technology and business strategy with a dynamic capabilities perspective: a longitudinal study of a Taiwanese semiconductor company. Int. J. Inf. Manag. **28**(5), 366–378 (2008)
10. Porter, M.E.: What is Strategy? Harvard Bus. Rev. **74**(6), 61–78 (1996)
11. Ciborra, U.C.: De Profundis? Deconstructing the concept of strategic alignment. Scandinavian J. Inf. Syst. **9**(1), 67–82 (1997)
12. Weill, P., Broadbent, M.: Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology. Harvard Business School Press, Boston (1998)
13. Luftman, J., Papp, R., Brier, T.: Business and IT Harmony: Enablers and Inhibitors to Alignment. http://Hsb.Baylor.Edu/Ramsower/AIS.Ac.96/Papers/PAPP.Htm (2000)
14. Henderson, J.C., Venkatraman, N.: Strategic Alignment: A Model for Organizational Transformation via Strategic Technology. Center for Information Systems Research, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Mass (1990)
15. Smaczny, T.: IS an alignment between business and IT the appropriate paradigm to manage IT in today's organization? Manag. Decis. J. **39**(10), 797–802 (2001)
16. Kitchenham, B.A., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature in software engineering – a systematic literature review. Inf. Softw. Technol. **51**, 7–15 (2009)

17. Pai, M., McCulloch, M., Gorman, J.D., Pai, N., Enanoria, W., Kennedy, G., Tharyan, P., Colford Jr., J.M.: Systematic reviews and meta-analyses: an illustrated, step-by-step guide. Nat. Med. J. India **17**(2), 84–95 (2004)
18. Carvalho, R., Sousa, P.: Business and Information Systems MisAlignment Model (BISMAM): an holistic model leveraged on misalignment and medical sciences approaches. In: Proceedings of BUSITAL (2008)
19. Papp, R.: Introduction to strategic alignment. In: Papp, R. (ed.) Strategic Information Technology: Opportunities for Competitive Advantage, pp. 1–24. Idea Group, Hershey (2001)
20. Pereira, C., Sousa, P.: Getting into the misalignment between business and information systems. In: Proceedings of 10th European Conference on Information Technology Evaluation (2003)
21. Avison, D., Jones, J., Powell, P., Wilson, D.: Using and validating the strategic alignment model. J. Strateg. Inf. Syst. **13**(3), 223–246 (2004)
22. Bleistein, S.J., Cox, K., Verner, J.: Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams. J. Syst. Softw. **79**(3), 362–378 (2006)
23. De Castro, V., Marcos, E., Vara, J.M.: Applying CIM-to-PIM model transformations for the service-oriented development of information systems. J. Inf. Softw. Technol. **53**(1), 87–105 (2011)
24. Aversano, L., Grasso, C., Tortorella, M.: Measuring the alignment between business processes and software systems: a case study. In: Proceedings of Symposium on Applied Computing (SAC'10), pp. 2330–2336. ACM press (2010)
25. Etien, A., Rolland, C.: Measuring the fitness relationship. Requir. Eng. J. **10**(3), 184–197 (2005)
26. Wieringa, R.J., Blanken, H.M., Fokkinga, M.M., Grefen, P.W.P.J.: Aligning Application Architecture to the Business Context. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 209–225. Springer, Heidelberg (2003)
27. Thevenet, L., Salinesi, C., Etien, A., Gam, I., Lasoued M.: Experimenting a modeling approach for designing organization's strategies in the context of strategic alignment. In: Proceedings of Australian Workshop on Requirements Engineering AWRE'06 (2006)
28. Chen, L.: Business–IT alignment maturity of companies in China. J. Inf. Manag. **47**(1), 9–16 (2010)
29. Hooper, V.A., Huff, S.L., Thirkell, P.C.: The impact of is-marketing alignment on marketing performance and business performance. Data Base **41**(1), 36–55 (2010)
30. Aversano, L., Marulli, F., Tortorella, M.: Recovering traceability links between business activities and software components. In: Quintela Varajão, J.E., Cruz-Cunha, M.M., Putnik, G.D., Trigo, A. (eds.) CENTERIS 2010, Part I. CCIS, vol. 109, pp. 385–394. Springer, Heidelberg (2010)
31. Reich, B., Benbasat, I.: Factors that influence the social dimension of alignment between business and information technology objectives. MIS Quart. **24**(1), 81–113 (2000)

Correction to: Time Efficiency of Point-of-Sale Payment Methods: Empirical Results for Cash, Cards and Mobile Payments

Michal Polasik, Jakub Górka, Gracjan Wilczewski,
Janusz Kunkowski, Karolina Przenajkowska, and Natalia Tetkowska

Correction to:
Chapter "Time Efficiency of Point-of-Sale Payment Methods: Empirical Results for Cash, Cards and Mobile Payments" in: J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, https://doi.org/10.1007/978-3-642-40654-6_19

The original version of this chapter, "Time Efficiency of Point-of-Sale Payment Methods: Empirical Results for Cash, Cards and Mobile Payments" was previously published as non-open access. It has now been changed to open access under a CC BY 4.0 license and the copyright holder has been updated as 'The Author(s)'. The book and the chapter have been updated with the change.

The updated version of this chapter can be found at
https://doi.org/10.1007/978-3-642-40654-6_19

J. Cordeiro et al. (Eds.): ICEIS 2012, LNBIP 141, pp. 1–1, 2013,
https://doi.org/10.1007/978-3-642-40654-6_29,

Author Index