

**Wilfred Ng
Veda C. Storey
Juan C. Trujillo (Eds.)**

LNCS 8217

Conceptual Modeling

**32th International Conference, ER 2013
Hong Kong, China, November 2013
Proceedings**



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Wilfred Ng Veda C. Storey
Juan C. Trujillo (Eds.)

Conceptual Modeling

32th International Conference, ER 2013
Hong Kong, China, November 11-13, 2013
Proceedings

Volume Editors

Wilfred Ng
The Hong Kong University of Science and Technology
Department of Computer Science and Engineering
Hong Kong, China
E-mail: wilfred@cse.ust.hk

Veda C. Storey
Georgia State University
J. Mack Robinson College of Business
Atlanta, GA, USA
E-mail: vstorey@gsu.edu

Juan C. Trujillo
University of Alicante, Spain
Department of Language and Information Systems
Alicante, Spain
E-mail: jtrujillo@dlsi.ua.es

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-41923-2 e-ISBN 978-3-642-41924-9
DOI 10.1007/978-3-642-41924-9
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: Applied for

CR Subject Classification (1998): D.2.1-4, I.6.5, F.3.2, H.2.4, H.2.7-8, H.3.3-4, I.2.4, H.4, J.1, K.6

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

© Springer-Verlag Berlin Heidelberg 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

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

Printed on acid-free paper

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

Foreword

This volume presents the papers selected for presentation at the 32nd International Conference on Conceptual Modeling (ER 2013), held in Hong Kong, November 11–13, 2013. The International Conference on Conceptual Modeling is the leading conference in the area of information systems and database design, attracting over 100 world-class researchers from around the world, who work in both academia and industry.

Following the tradition of the conferences, ER 2013 provided a forum to exchange ideas and experiences, and to discuss current research and applications, with a major focus on conceptual modeling. The conference topics include: theories of concepts and ontologies underlying conceptual modeling, methods and tools for developing and communicating conceptual models, and techniques for transforming conceptual models into effective implementations. This year, the conference was held in the greater Shatin area of Hong Kong, which features such places of interest as the Ten Thousand Buddhas Monastery, the Hong Kong Heritage Museum, and the Shatin Racecourse. On the banks of the Shing Mun River, the event was located within a cluster of malls and has easy connections to many transportation links such as the MTR (Hong Kong subway system). The nearby New Town Plaza offers an immense variety of brand boutique as well as dining and entertainment venues.

ER 2013 was the outcome of the joint effort of many sponsors, colleagues, students, and volunteers. In particular, it was one of the celebratory events of the 50th Anniversary of The Chinese University of Hong Kong. We wish to express our gratitude for the help and contributions from other sponsors, including the City University of Hong Kong, the K.C. Wong Education Foundation, and the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology. We would also like to thank the authors who submitted papers, demos, and tutorials, and panels as well as the conference participants. We are especially grateful to the ER Steering Committee members for their support in organizing ER 2013, and to the chairs and members of the Technical Program Committee and the external reviewers for their relentless work in reviewing submissions with expertise and patience in order to develop a top-quality scientific program. Many thanks as well to the Workshop, Panel, Tutorial, Demos, PhD Symposium Chairs and committee members for their professional services. ER 2013 owes special thanks to the organizers of the Symposium of Conceptual Modeling Education, a new initiative to this conference series, and to the Local Organization and the Fund-raising Chairs, who

devoted a great deal of time and energy to ensure the success of this conference. Thank you to our Publicity Chairs, our Webmaster, and our Registration and Proceedings Chairs, for their work in ushering this conference through its various stages of development.

July 2013

Qing Li
Ho-fung Leung

Preface

Since the first version of the entity-relationship (ER) model by Peter Chen appeared in *ACM Transactions on Database Systems (TODS)* in 1976, both the ER model and conceptual modeling have been key success factors for modeling computer-based systems. The International Conference on Conceptual Modeling is an important venue for the presentation and exchange of ideas and concepts that relate to traditional and emerging issues in conceptual modeling of information systems. Work on conceptual modeling has continued to evolve as the ER model has been applied, modified, and extended to research in database management systems, business process management, and management information systems. Conceptual modeling plays a vital role in the emerging, new data era where the correct design and development of mobile or sensors analytics, big data systems, non-SQL databases, smart cities and biomedical systems will be crucial. The 32nd International Conference on conceptual modeling was a forum where some of these novel areas, as well as their fundamental and theoretical issues which are directly related to conceptual modeling, were discussed.

The ER conference continues to attract some of the best researchers and keynote speakers, from both academia and industry, who work on topics in traditional and emerging areas of conceptual modeling. This year, 148 abstracts and 126 full papers were submitted to the conference. Each paper was reviewed by at least three reviewers and, based upon these reviews, 23 full papers and 17 short papers were selected for publication in the proceedings and presentation at the conference. The acceptance rate for regular papers was 18.25%, and for regular and short papers together, 31.74%. These papers were organized into sessions that represent leading research areas in conceptual modeling, including topics related to querying, semantics, fundamental concepts, applications, and emerging issues. The program included four research prototype demos providing an interactive way for participants to appreciate contemporary issues in conceptual modeling research. The demos have prior, corresponding theoretical publications. The four demos address different conceptual modeling issues related to implementations, applications, and innovative techniques, thus making visible the pragmatic aspects of conceptual modeling. The scientific program also featured three interesting keynote presentations by David Embley, Marie-Aude Aufaure, and Surajit Chaudhuri, each of whom has shared some of their thoughts and insights in these proceedings.

We wish to thank the 101 Program Committee members and the external reviewers who provided insightful reviews and discussions on the papers. We also appreciate the diligence of the senior reviewers who provided guidance and

recommendations, as well the selection of best paper awards. Most importantly, we thank the authors who submitted high-quality research papers on a wide variety of topics, thus making this conference possible. We hope you enjoy the program.

July 2013

Wilfred Ng
Veda C. Storey
Juan C. Trujillo

Conference Organization

Honorary Chairs

Peter P. Chen Carnegie Mellon University and LSU, USA
J. Leon Zhao City University of Hong Kong, Hong Kong

Conference Co-chairs

Qing Li City University of Hong Kong, Hong Kong
Ho-fung Leung Chinese University of Hong Kong, Hong Kong

Technical Program Co-chairs

Wilfred Ng Hong Kong University of Science and
Technology, Hong Kong
Juan Trujillo University of Alicante, Spain
Veda Storey Georgia State University, USA

Workshop Co-chairs

Jeffrey Parsons Memorial University of Newfoundland, Canada
Dickson Chiu Dickson Computer Systems, Hong Kong

Publicity Co-chairs

Sandeep Purao Penn State University, USA
Dongqing Yang Peking University, China

Panel Chairs

Heinrich C. May University of Klagenfurt, Austria
Sudha Ram University of Arizona, USA

Tutorial Chairs

Antoni Olive Universitat Politècnica de Catalunya, Spain
Zhiyong Peng Wuhan University, China

Poster/Demo Chairs

Carson Woo	UBC, Canada
Kamal Karlapalem	IIIT, India

PhD Symposium Co-chairs

Stephen Liddle	Brigham Young University, USA
Mengchi Liu	Carleton University, Canada

Educational Symposium Co-chairs

James Cheng	Chinese University of Hong Kong, Hong Kong
Il-Yeol Song	Drexel University, USA

Local Organization Chairs

Jean Wang	City University of Hong Kong, Hong Kong
Hong Va Leong	Hong Kong Polytechnic University, Hong Kong

Fund Raising Chair

Kam-Fai Wong	Chinese University of Hong Kong, Hong Kong
--------------	--

Finance Chair

Howard Leung	City University of Hong Kong, Hong Kong
--------------	---

Webmaster

Xudong Mao	City University of Hong Kong, Hong Kong
------------	---

Steering Committee Liaison

Tok Wang Ling	National University of Singapore, Singapore
---------------	---

Organized By

City University of Hong Kong
Chinese University of Hong Kong

Sponsored By

K.C. Wong Education Foundation
City University of Hong Kong
Chinese University of Hong Kong
Hong Kong University of Science and Technology

In cooperation with

ACM Hong Kong Chapter
IEEE Computer Society Hong Kong Chapter
Hong Kong Web Society

Program Committee

Jacky Akoka	CNAM and TEM, France
Yuan An	Drexel University, USA
Manish Anand	Salesforce.com
Joao Araujo	Universidade Nova de Lisboa, Portugal
Akhilesh Bajaj	University of Tulsa, USA
Zhifeng Bao	University of Singapore, Singapore
Palash Bera	Saint Louis University, USA
Rafael Berlanga	Universitat Jaume I, Spain
Sandro Bimonte	Irstea, France
Shawn Bowers	Gonzaga University, USA
Stephane Bressan	National University of Singapore, Singapore
Jordi Cabot	Inria-École des Mines de Nantes, France
Silvana Castano	University of Milan, Italy
Jaelson Castro	UFPE - Universidade Federal de Pernambuco, Brazil
Stefano Ceri	DEI, Politecnico di Milano, Italy
James Cheng	Chinese University of Hong Kong, SAR China
Roger Chiang	AIS
Isabel F. Cruz	University of Illinois, USA
Faiz Currim	University of Arizona, USA
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria, Italy
Karen Davis	University of Cincinnati, USA
Umeshwar Dayal	Hewlett-Packard
Valeria De Antonellis	University of Brescia, Italy
José Palazzo M. De Oliveira	UFRGS, Brazil
Lois Delcambre	Portland State University, USA
Gill Dobbie	University of Auckland, New Zealand
David W. Embley	Brigham Young University, USA
Joerg Evermann	Memorial University of Newfoundland, Canada
Xavier Franch	Universitat Politècnica de Catalunya, Spain

Avigdor Gal	Technion, Israel
Felix Garcia	University of Castilla-La Mancha, Spain
Paolo Giorgini	University of Trento, Italy
Giancarlo Guizzardi	Federal University of Espirito Santo (UFES), Brazil
Jean-Luc Hainaut	University of Namur, Belgium
Arantza Illarramendi	Basque Country University, Spain
Hemant Jain	University of Wisconsin-Milwaukee, USA
Matthias Jarke	RWTH Aachen University, Germany
Manfred Jeusfeld	Tilburg University, The Netherlands
Ivan Jureta	University of Namur, Belgium
Larry Kerschberg	George Mason University, USA
Vijay Khatri	Indiana University, USA
Alberto Laender	Federal University of Minas Gerais, Brazil
Dik Lun Lee	Hong Kong University of Science and Technology, SAR China
Mong Li Lee	National University of Singapore, Singapore
Sang-Goo Lee	Seoul National University, Korea
Wolfgang Lehner	TU Dresden, Germany
Julio Cesar Leite	PUC-Rio, Brazil
Stephen Liddle	Brigham Young University, USA
Tokwang Ling	National University of Singapore, Singapore
Mengchi Liu	Carleton University, Canada
Fred Lochovsky	HKUST, SAR China
Oscar P. Lopez	PROS Research Centre, Universitat Politècnica de València, Spain
Pericles Loucopoulos	Loughborough University, UK
Wolfgang Maass	Saarland University, Germany
Patrick Marcel	Université François Rabelais Tours, Laboratoire d'Informatique, France
Esperanza Marcos	Universidad Rey Juan Carlos, Spain
Heinrich C. Mayr	Alpen-Adria-Universität Klagenfurt, Austria
Jan Mendling	Wirtschaftsuniversität Wien, Austria
Ana Moreira	Universidade Nova de Lisboa, Portugal
John Mylopoulos	University of Toronto, Canada
Miyuki Nakano	University of Tokyo, Japan
Antoni Olivé	Universitat Politècnica de Catalunya, Spain
Andreas L. Opdahl	University of Bergen, Norway
Jeffrey Parsons	Memorial University of Newfoundland, Canada
Norman Paton	The University of Manchester, UK
Zhiyong Peng	State Key Lab. of Software Engineering, China
Barbara Pernici	Politecnico di Milano, Italy
Geert Poels	Ghent University, Belgium
Henderik Proper	Public Research Centre Henri Tudor, Luxembourg

Sandeep Purao	The Pennsylvania State University, USA
Christoph Quix	RWTH Aachen University, Germany
Jolita Ralyté	University of Geneva, Switzerland
Sudha Ram	University of Arizona, USA
Maryam Razavian	VU University Amsterdam, The Netherlands
Iris Reinhartz-Berger	University of Haifa, Israel
Stefano Rizzi	DEIS - University of Bologna, Italy
Colette Rolland	Université Paris 1 Panthéon - Sorbonne, France
Antonio Ruiz	University of Seville, Spain
Mehrdad Sabetzadeh	University of Luxembourg, Luxembourg
Motoshi Saeki	Tokyo Institute of Technology, Japan
Satya Sahoo	Case Western Reserve University, USA
Camille Salinesi	Université Paris 1 Panthéon - Sorbonne, France
Peretz Shoval	Ben-Gurion University, Israel
Il-Yeol Song	Drexel University, USA
Vijayan Sugumaran	Oakland University, USA
David Taniar	Monash University, Australia
Ernest Teniente	Universitat Politècnica de Catalunya, Spain
James Terwilliger	Microsoft Corporation
Bernhard Thalheim	University of Kiel, Germany
Panos Vassiliadis	University of Ioannina, Greece
Ramesh Venkataraman	Indiana University, USA
Gerd Wagner	Brandenburg University of Technology at Cottbus, Germany
Yair Wand	UBC, Canada
Xiaoling Wang	Fudan University, China
Barbara Weber	University of Innsbruck, Austria
Roel Wieringa	University of Twente, The Netherlands
Carson Woo	University of British Columbia, Canada
Huayu Wu	Institute for Infocomm Research, Singapore
Masatoshi Yoshikawa	Nagoya University, Japan
Eric Yu	University of Toronto, Canada
Zhu Zhang	University of Arizona, USA
Huimin Zhao	University of Wisconsin-Milwaukee, USA
Shuigeng Zhou	Fudan University, China
Esteban Zimányi	Université Libre de Bruxelles, Belgium

Additional Reviewers

Ahn, Yeonchan	Bianchini, Devis
Alencar, Fernanda	Bjeković, Marija
Aligon, Julien	Busany, Nimrod
Ayala, Claudia P.	Cabanillas, Cristina
Bermudez, Jesus	Cappiello, Cinzia

Chen, Shixi
Costal, Dolors
Danesh, Mohammad
Dannecker, Lars
Del Río Ortega, Adela
Feltus, Christophe
Fugini, Mariagrazia
Huang, Silu
Lopez-Sanz, Marcos
Lukyanenko, Roman
Martínez, Salvador
Melchiori, Michele
Mena, Eduardo
Molnar, Wolfgang
Nicolae, Oana
Oliveira, Karolyne
Park, Youngki
Pimentel, João
Pinet, François
Plebani, Pierluigi
Pérez, María
Raad, Elie
Resinas, Manuel
Rull, Guillem
Sanz, Ismael
Sekhavat, Yoones A.
Shekhovtsov, Vladimir A.
Silva, Carla
Sánchez Fúquene, Diana Marcela
Sörensen, Ove
Trinidad, Pablo
Vara, Juan Manuel
Wu, Huanhuan
Wu, Wei
Xu, Yanyan
Yang, Zhenglu
Yeon, Jonghm

Sponsors



王寬誠教育基金會
K. C. WONG EDUCATION FOUNDATION



Keynotes

(Abstracts)

Big Data—Conceptual Modeling to the Rescue

David W. Embley¹ and Stephen W. Liddle²

¹ Department of Computer Science

² Information Systems Department

Brigham Young University, Provo, Utah 84602, USA

embley@cs.byu.edu, liddle@byu.edu

Abstract. Big data is characterized by volume, variety, velocity, and veracity. We should expect conceptual modeling to provide some answers since its historical perspective has always been about structuring information—making its volume searchable, harnessing its variety uniformly, mitigating its velocity with automation, and checking its veracity with application constraints. We provide perspectives about how conceptual modeling can “come to the rescue” for many big-data applications by handling volume and velocity with automation, by inter-conceptual-model transformations for mitigating variety, and by conceptualized constraint checking for increasing veracity.

What's Up in Business Intelligence? A Contextual and Knowledge-Based Perspective

Marie-Aude Auaure

Ecole Centrale Paris, MAS Laboratory, Chatenay-Malabry, France
Marie-Aude.Aufaure@ecp.fr

Abstract. The explosive growth in the amount of data poses challenges in analyzing large data sets and retrieving relevant information in real-time. This issue has dramatically increased the need for tools that effectively provide users with means of identifying and understanding relevant information. Business Intelligence (BI) promises the capability of collecting and analyzing internal and external data to generate knowledge and value, providing decision support at the strategic, tactical, and operational levels. Business Intelligence is now impacted by the Big Data phenomena and the evolution of society and users, and needs to take into account high-level semantics, reasoning about unstructured and structured data, and to provide a simplified access and better understanding of data. This paper will depict five years research of our academic chair in Business Intelligence from the data level to the user level, mainly focusing on the conceptual and knowledge level.

Big Data and Enterprise Analytics

Surajit Chaudhuri

Microsoft Research, USA

surajitc@microsoft.com

Abstract. In this talk, I will describe the key secular trends that characterize the field of Big Data with respect to enterprise analytics. I will describe some of the open challenges for enterprise analytics in the context of Big Data. Although some of these problems are not new, their importance is amplified by Big Data. As an example, we will discuss the task of data exploration and leveraging unstructured data for enterprise analytics.

Table of Contents

Keynotes

- Big Data—Conceptual Modeling to the Rescue 1
David W. Embley and Stephen W. Liddle

- Whats Up in Business Intelligence? A Contextual and Knowledge-Based Perspective 9
Marie-Aude Aufaure

Modeling and Reasoning

- Modeling and Reasoning with Decision-Theoretic Goals 19
Sotirios Liaskos, Shakil M. Khan, Mikhail Soutchanski, and John Mylopoulos

- TBIM: A Language for Modeling and Reasoning about Business Plans 33
Fabiano Francesconi, Fabiano Dalpiaz, and John Mylopoulos

- Automated Reasoning for Regulatory Compliance 47
Alberto Siena, Silvia Ingolfo, Anna Perini, Angelo Susi, and John Mylopoulos

Fundamentals of Conceptual Modeling

- Is Traditional Conceptual Modeling Becoming Obsolete? 61
Roman Lukyanenko and Jeffrey Parsons

- Cognitive Mechanisms of Conceptual Modelling: How Do People Do It? 74
Ilona Wilmont, Sytse Hengeveld, Erik Barendsen, and Stijn Hoppenbrouwers

- A Semantic Analysis of Shared References 88
Roland Kaschek

- Are Conceptual Models Concept Models? 96
Chris Partridge, Cesar Gonzalez-Perez, and Brian Henderson-Sellers

Business Process Modeling I

Visual Modeling of Business Process Compliance Rules with the Support of Multiple Perspectives	106
<i>David Knuplesch, Manfred Reichert, Linh Thao Ly, Akhil Kumar, and Stefanie Rinderle-Ma</i>	
Deciding Data Object Relevance for Business Process Model Abstraction	121
<i>Josefine Harzmann, Andreas Meyer, and Mathias Weske</i>	
Matching Business Process Models Using Positional Passage-Based Language Models	130
<i>Matthias Weidlich, Eitam Sheetrit, Moisés C. Branco, and Avigdor Gal</i>	
Towards an Empirically Grounded Conceptual Model for Business Process Compliance	138
<i>Martin Schultz</i>	

Business Process Modeling II

Improving Business Process Intelligence with Object State Transition Events	146
<i>Nico Herzberg, Andreas Meyer, Oleh Khovalko, and Mathias Weske</i>	
A Conceptual Model of Intended Learning Outcomes Supporting Curriculum Development	161
<i>Preecha Tangworakitthaworn, Lester Gilbert, and Gary B. Wills</i>	
Cost-Informed Operational Process Support	174
<i>Moe T. Wynn, Hajo A. Reijers, Michael Adams, Chun Ouyang, Arthur H.M. ter Hofstede, Wil M.P. van der Aalst, Michael Rosemann, and Zahirul Hoque</i>	

Network Modeling

Automating the Adaptation of Evolving Data-Intensive Ecosystems	182
<i>Petros Manousis, Panos Vassiliadis, and George Papastefanatos</i>	
sonSchema: A Conceptual Schema for Social Networks	197
<i>Zhifeng Bao, Y.C. Tay, and Jingbo Zhou</i>	
Minimizing Human Effort in Reconciling Match Networks	212
<i>Hung Quoc Viet Nguyen, Tri Kurniawan Wijaya, Zoltán Miklós, Karl Aberer, Eliezer Levy, Victor Shafran, Avigdor Gal, and Matthias Weidlich</i>	

Data Semantics

Effective Recognition and Visualization of Semantic Requirements by Perfect SQL Samples	227
<i>Van Bao Tran Le, Sebastian Link, and Flavio Ferrarotti</i>	
A Semantic Approach to Keyword Search over Relational Databases	241
<i>Zhong Zeng, Zhifeng Bao, Mong Li Lee, and Tok Wang Ling</i>	
Semantic-Based Mappings	255
<i>Giansalvatore Mecca, Guillem Rull, Donatello Santoro, and Ernest Teniente</i>	

Security and Optimization

Managing Security Requirements Conflicts in Socio-Technical Systems	270
<i>Elda Paja, Fabiano Dalpiaz, and Paolo Giorgini</i>	
Optimising Conceptual Data Models through Profiling in Object Databases	284
<i>Tilmann Zäschke, Stefania Leone, Tobias Gmünder, and Moira C. Norrie</i>	
Skyline Queries over Incomplete Data - Error Models for Focused Crowd-Sourcing	298
<i>Christoph Lofi, Kinda El Maarry, and Wolf-Tilo Balke</i>	

Ontology-Based Modeling I

Toward an Ontology-Driven Unifying Metamodel for UML Class Diagrams, EER, and ORM2	313
<i>C. Maria Keet and Pablo Rubén Fillottrani</i>	
Towards Ontological Foundations for the Conceptual Modeling of Events	327
<i>Giancarlo Guizzardi, Gerd Wagner, Ricardo de Almeida Falbo, Renata S.S. Guizzardi, and João Paulo A. Almeida</i>	
Baquara: A Holistic Ontological Framework for Movement Analysis Using Linked Data	342
<i>Renato Fileto, Marcelo Krüger, Nikos Pelekis, Yannis Theodoridis, and Chiara Renso</i>	

Searching and Mining

From Structure-Based to Semantics-Based: Towards Effective XML Keyword Search	356
<i>Thuy Ngoc Le, Huayu Wu, Tok Wang Ling, Luochen Li, and Jiaheng Lu</i>	
Combining Personalization and Groupization to Enhance Web Search	372
<i>Kenneth Wai-Ting Leung, Dik Lun Lee, and Yuchen Liu</i>	

Colored Petri Nets for Integrating the Data Perspective in Process Audits	387
<i>Michael Werner</i>	

Conceptual Modeling and Applications I

Former Students' Perception of Improvement Potential of Conceptual Modeling in Practice	395
<i>Albert Tort, Antoni Olivé, and Joan Antoni Pastor</i>	

Conceptual Modeling for Ambient Assistance	403
<i>Judith Michael and Heinrich C. Mayr</i>	

Empirical Evaluation of the Quality of Conceptual Models Based on User Perceptions: A Case Study in the Transport Domain	414
<i>Daniela S. Cruzes, Audun Vennesland, and Marit K. Natvig</i>	

Conceptual Modeling and Applications II

Towards the Effective Use of Traceability in Model-Driven Engineering Projects	429
<i>Iván Santiago, Juan Manuel Vara, María Valeria de Castro, and Esperanza Marcos</i>	

Modeling Citizen-Centric Services in Smart Cities	438
<i>Sandeep Purao, Teo Chin Seng, and Alfred Wu</i>	

Representing and Elaborating Quality Requirements: The QRA Approach	446
<i>Jie Sun, Pericles Loucopoulos, and Liping Zhao</i>	

Towards a Strategy-Oriented Value Modeling Language: Identifying Strategic Elements of the VDML Meta-model	454
<i>Ben Roelens and Geert Poels</i>	

Ontology-Based Modeling II

Ontological Distinctions between Means-End and Contribution Links in the i* Framework	463
<i>Renata S.S. Guizzardi, Xavier Franch, Giancarlo Guizzardi, and Roel Wieringa</i>	
Applying the Principles of an Ontology-Based Approach to a Conceptual Schema of Human Genome	471
<i>Ana M^a Martínez Ferrandis, Oscar Pastor López, and Giancarlo Guizzardi</i>	
Ontologies for International Standards for Software Engineering.....	479
<i>Brian Henderson-Sellers, Tom McBride, Graham Low, and Cesar Gonzalez-Perez</i>	
On the Symbiosis between Enterprise Modelling and Ontology Engineering	487
<i>Frederik Gailly, Sven Casteleyn, and Nadejda Alkhaldi</i>	
 Demonstration Papers	
sonSQL: An Extensible Relational DBMS for Social Network Start-Ups	495
<i>Zhifeng Bao, Jingbo Zhou, and Y.C. Tay</i>	
OntoDBench: Interactively Benchmarking Ontology Storage in a Database	499
<i>Stéphane Jean, Ladjel Bellatreche, Carlos Ordonez, Géraud Fokou, and Mickaël Baron</i>	
Specifying and Reasoning over Socio-Technical Security Requirements with STS-Tool	504
<i>Elda Paja, Fabiano Dalpiaz, Mauro Poggianella, Pierluigi Roberti, and Paolo Giorgini</i>	
Lightweight Conceptual Modeling for Crowdsourcing	508
<i>Roman Lukyanenko and Jeffrey Parsons</i>	
Author Index	513

Big Data—Conceptual Modeling to the Rescue

David W. Embley¹ and Stephen W. Liddle²

¹ Department of Computer Science

² Information Systems Department

Brigham Young University, Provo, Utah 84602, USA

embley@cs.byu.edu, liddle@byu.edu

Abstract. Big data is characterized by volume, variety, velocity, and veracity. We should expect conceptual modeling to provide some answers since its historical perspective has always been about structuring information—making its volume searchable, harnessing its variety uniformly, mitigating its velocity with automation, and checking its veracity with application constraints. We provide perspectives about how conceptual modeling can “come to the rescue” for many big-data applications by handling volume and velocity with automation, by inter-conceptual-model transformations for mitigating variety, and by conceptualized constraint checking for increasing veracity.

Keywords: Big Data, Conceptual Modeling.

1 Big Data

Every day humans generate several petabytes of data [1] from a variety of sources such as orbital weather satellites, ground-based sensor networks, mobile computing devices, digital cameras, and retail point-of-sale registers. Companies, governments, and individuals store this data in a wide variety of structured, semistructured, and unstructured formats. However, most of this data either languishes in underutilized storage repositories or is never stored in the first place. Ironically, in an era of unprecedented access to a veritable gold mine of information, it is increasingly difficult to unlock the value stored within our data. The essential problem of “Big Data” is that we are accumulating data faster than we can process it, and this trend is accelerating.

The so-called “four V’s” characterize Big Data:

- *Volume*: applications sometimes exceeding petabytes¹
- *Variety*: widely varying heterogeneous information sources and hugely diverse application needs

¹ Having successfully communicated the terms “mega-,” “giga-,” and “tera-byte,” in the Big Data era we now need to teach users about “peta-,” “exa-,” “zetta-,” and even “yotta-bytes.” The NSA data center being built in Utah within 35km of our university purportedly is designed to store at least zettabytes (10^{21} bytes) and perhaps yottabytes (10^{24} bytes) of data.

- *Velocity*: phenomenal rate of data acquisition, real-time streaming data, and variable time-value of data
- *Veracity*: trustworthiness and uncertainty, beyond the limits of humans to check

We should expect conceptual modeling to provide some answers since its historical perspective has always been about structuring information—making its *volume* searchable, harnessing its *variety* uniformly, mitigating its *velocity* with automation, and checking its *veracity* with application constraints. We do not envision any silver bullets that will slay the “werewolf” of Big Data, but conceptual modeling can help, as we illustrate with an example from our project that seeks to superimpose a web of knowledge over a rapidly growing heterogeneous collection of historical documents whose storage requirements are likely to eventually exceed many exabytes.

2 Conceptual Modeling to the Rescue

Can conceptual modeling “come to the rescue” in some sense and help address some of the challenges of Big Data? We believe that the answer is affirmative. We do not expect conceptual modeling to address such issues as how we physically store and process bits of data, but “Moore’s Law”² gives us confidence that future hardware technology will also help address these challenges. The mapping of conceptual models to efficient storage structures, a traditional application of conceptual modeling, is likely to be vastly different for Big Data and may be of some interest. But logical-to-physical design is not where we see the impact of conceptual modeling on Big Data. We expect that conceptual modeling can help by conceptual-model-based extraction for handling *volume* and *velocity* with automation, by inter-conceptual-model transformations for mitigating *variety*, and by conceptualized constraint checking for increasing *veracity*.

Consider the application of family-history information as captured in historical books. We have access to a collection of 85,000 such books that describe family genealogy, biographies, family stories, photos, and related information. These documents contain a variety of pages as Figures 1 and 2 illustrate. Documents such as these are information-dense, containing many assertions both directly stated and implied. For example, from the page in Figure 1 we read that Mary Ely was born to Abigail Huntington Lathrop in 1838—the author stated this assertion directly. However we also can infer that Mary was a daughter of her mother Abigail because “Mary” and “Abigail” are generally accepted as a female names. This type of information—including both stated and inferred assertions—is useful to someone who is searching for information about members of this family.

² Moore’s Law is not strictly speaking a law, but rather Gordon Moore’s observation that the number of transistors on an integrated circuit doubles approximately every two years. The observation has generally held true since 1965, though some observers believe the rate of growth will soon decrease. See http://en.wikipedia.org/wiki/Moore's_law

THE ELY ANCESTRY.
SEVENTH GENERATION.

419

241213. Mary Eliza Warner, b. 1826, dau. of Samuel Selden Warner and Azubah Tully; m. 1850, Joel M. Gloyd (who was connected with Chief Justice Waite's family).

243311. Abigail Huntington Lathrop (widow), Boonton, N. J., b. 1810, dau. of Mary Ely and Gerard Lathrop; m. 1835, Donald McKenzie, West Indies, who was b. 1812, d. 1839.

(The widow is unable to give the names of her husband's parents.)
Their children:

- 1. Mary Ely, b. 1836, d. 1859.
- 2. Gerard Lathrop, b. 1838.

243312. William Gerard Lathrop, Boonton, N. J., b. 1812, d. 1882, son of Mary Ely and Gerard Lathrop; m. 1837, Charlotte Brackett Jennings, New York City, who was b. 1818, dau. of Nathan Tilestone Jennings and Maria Miller. Their children:

- 1. Maria Jennings, b. 1838, d. 1840.
- 2. William Gerard, b. 1840.
- 3. Donald McKenzie, b. 1840, d. 1843. } Twins.
- 4. Anna Margareta, b. 1843.
- 5. Anna Catherine, b. 1845.

243314. Charles Christopher Lathrop, N. Y. City, b. 1817, d. 1865, son of Mary Ely and Gerard Lathrop; m. 1856, Mary Augusta Andruss, 992 Broad St., Newark, N. J., who was b. 1825, dau. of Judge Caleb Halstead Andruss and Emma Sutherland Goble. Mrs. Lathrop died at her home, 992 Broad St., Newark, N. J., Friday morning, Nov. 4, 1898. The funeral services were held at her residence on Monday, Nov. 7, 1898, at half-past two o'clock P. M. Their children:

- 1. Charles Halstead, b. 1857, d. 1861.
- 2. William Gerard, b. 1858, d. 1861.
- 3. Theodore Andruss, b. 1860.
- 4. Emma Goble, b. 1862.

Miss Emma Goble Lathrop, official historian of the New York Chapter of the Daughters of the American Revolution, is one of the youngest members to hold office, but one whose intelligence and capability qualify her for such distinction. Miss Lathrop is not without experience; in her present home and native city, Newark, N. J., she has filled the positions of secretary and treasurer to the Girls' Friendly Society for nine years, secretary and president of the Woman's Auxiliary of Trinity Church Parish, treasurer of the St. Catherine's Guild of St. Barnabas Hospital, and manager of several of Newark's charitable institutions which her grandparents were instrumental in founding. Miss Lathrop traces her lineage back through many generations of famous progenitors on both sides. Her maternal ancestors were among the early settlers of New Jersey, among them John Ogden, who received patent in 1664 for the purchase of Elizabethtown, and who in 1673 was

Page 3 —



The Ashman house on the hill where Dale was born, fall of 1910.

Grandma had hoped Dale would be a girl. After he was born, her husband accused her of making a girl out of this golden, curly haired boy. She loved fixing his hair in long ringlets but when he was four he'd had enough of that and wanted his hair cut. Grandpa took him to the barber but when he was seated in the chair with the towel around his neck, he became fearful. Grandfather Ashman could never stand to see one of his children cry so he took Dale across the street and bought him a small box of fairy stick candy.

Next they went to a nearby photography shop and Dale had his photo taken. Then they went back to face the barber. This time Dale let him cut his ringlets. According to Uncle Harold when they returned home from the barber Grandma took one look at her little boy and began to sob.

Fig. 1. Ely Ancestry page

Fig. 2. Dale Ashman page

Assume that each book has approximately 500 pages, that there are 100 stated and 100 inferred assertions per page, and that each assertion requires 500 bytes of storage. Further assume that each page needs to be stored both as a high-resolution image and as a processed textual representation, taking another 10,000 and 1,000 bytes respectively.³ We conservatively estimate that we could store a fact base extracted from these 85,000 documents in $85,000 \times 500 \times ((100 + 100) \times 500 + 10,000 + 1,000) = 4,717,500,000,000$ bytes. The 4.7 terabytes constitutes a modestly large data store, though fairly manageable, and we guess that compression techniques could reduce the storage requirement into the sub-terabyte range.

However, this collection is only the beginning within the family-history application domain. There are many such collections of historical family-history books, and a variety of other related information sources of interest, both static

³ These assumptions are based on approximate averages we have observed in our actual work on historical documents.

Ontology Snippets:

ChildRecord

external representation: $\wedge(\backslash d\{1,3\})\backslash.\backslash s+([A-Z]\backslash w+\backslash s[A-Z]\backslash w+)$
 $(,\backslash sb\backslash.\backslash s([1][6-9]\backslash d\backslash d))?(,\backslash sd\backslash.\backslash s([1][6-9]\backslash d\backslash d))?\backslash.$

predicate mappings: *Child(x); Child-ChildNr(x,1); Person-Name(x,2);*
Person-BirthDate(x,4); Person-DeathDate(x,6)

Fig. 3. Ontology Snippet Example. (The **predicate mappings** associate the text recognized by the regular-expression capture groups 1, 2, 4, and 6 with new child *x* in their respective relationships in Figure 4.)

and dynamic. For example, census records, ship manifests, historical newspapers, parish records, and military records are just a few of the types of information that a family-history company like Ancestry.com is interested in gathering, integrating, and making available to its clients. In addition to static sources, there are also dynamic sources such as family blogs, shared photo albums, and the Facebook social graph that could usefully augment the historical document base. Taken together, these sources easily exceed many exabytes of data. So the family-history domain certainly exhibits the *volume*, *variety*, and *velocity* challenges characteristic of Big Data. This domain also expresses the *veracity* dimension: it is common for multiple sources to make conflicting assertions about family-history details such as dates, places, names, person identity, and familial relationships.

Returning now to the relatively modest collection of 85,000 historical books, it is true that a search engine such as Lucene⁴ could readily be used to construct a full-text keyword index of this document base. However, keyword search, while a good start, is not nearly enough to accomplish the types of semantic searches we need. Is it possible to apply semantic markup to the concepts contained within those pages, semantically index the information for search and query, and clean and organize it as a valuable storage repository? Manually, with crowdsourcing, this may be possible, but neither the expense nor the timeliness would be tolerable. We see several ways conceptual modeling can “come to the rescue” to enable this application—and, by implication, to enable similar applications:

- *Conceptual-Model-Based Information Extraction.* To address the Big Data issues of *volume*, *velocity*, and *variety*, we take a conceptual-modeling approach to information extraction. We create a conceptual model that conforms to an author’s point of view and linguistically ground the conceptual model, turning it into an extraction ontology [2,3]. We linguistically ground a conceptual model by associating regular-expression pattern recognizers with the object and relationship sets of the conceptual model or with coherent collections of object and relationship sets, which we call ontology snippets. For example, we can declare the ontology snippet in Figure 3 to extraction information into the conceptual model in Figure 4, which represents the author’s view of the child lists in Figure 1. Further, since manual creation of

⁴ See <http://lucene.apache.org>

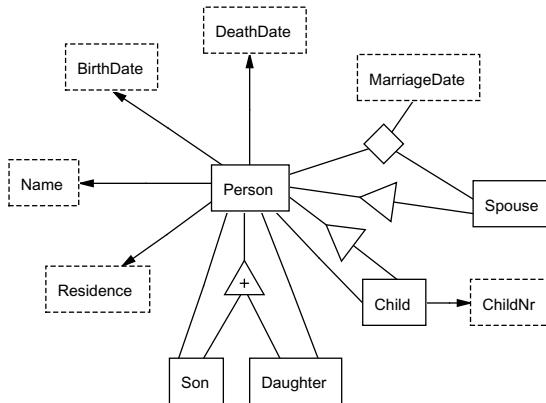


Fig. 4. Ontological Conceptualization for Assertion Extraction

pattern recognizers is likely to be too expensive for the *volume*, *velocity*, and *variety* of Big Data applications, we seek for ways to automatically generate recognizers (see [4] for an example).

- *Conceptual-Model-Based Knowledge Organization.* Information that is extracted with respect to an author’s view is often not ideally organized for search and query. Moreover, we are often interested not only in the stated assertions that can be extracted but also in what can be inferred from the stated assertions. Figure 5 shows a conceptualization of the way we may wish to organize the information in Figure 4 or the information extracted from any other historical document containing family-history information. Because conceptual models are or can be formally based on predicate calculus, we can use inference rules that map from one conceptual model to another to organize our knowledge base. For example, we can reorganize the *Son* and *Daughter* information in Figure 4 as *Child* information in Figure 5 and the *Name* as a multi-token string into an aggregate of *GivenNames* and a *Surname*. We can also infer *Gender*, which is almost never stated explicitly, either from the *Son* and *Daughter* classification or from *GivenNames* based on a probabilistic model of male and female names in nineteenth century America. (See [5] for an explanation about how we use Jena⁵ inference rules to map one conceptualization to another.) Besides conceptual organization, we would also like to resolve object identity. Of the four mentions of the name “Mary Ely” in Figure 1, three denote the same person, but the “Mary Ely” who is the daughter of Abigail is certainly different since she is the granddaughter of the other Mary Ely. We take a conceptual-modeling approach to resolving object identity. We extract and organize facts and then check, for example, whether two people with similar names have the same parents or were born in the same location on the same date. (See [5] for an

⁵ <http://jena.apache.org/>

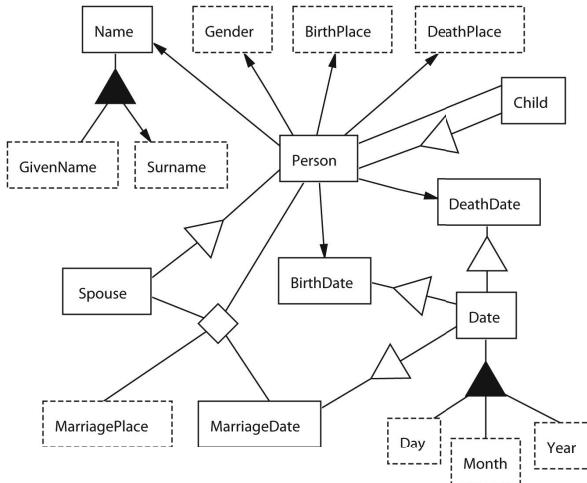


Fig. 5. Target Ontology of Desired Biographical Assertions

explanation about how we use the Duke⁶ entity resolution tool to resolve object identity.)

- *Conceptual-Model-Based Semantic Indexing and Query Processing.* To support the unlocking of the “veritable gold mine of information” in Big Data applications, we provide a conceptual-model-based, semantic-search mechanism that includes semantic indexing, free-form and advanced form-based query processing, and cross-language query processing:

- Semantic Indexing. To answer queries quickly, we must semantically crawl and index resources in advance. To create semantic indexes, we apply conceptual-model-based extraction ontologies to resources; we also pre-execute inference rules so that we index not only stated assertions but also inferred assertions [6].
- Free-form Query Processing. We process free-form queries in a hybrid fashion. We first apply extraction ontologies to a query to discover any semantics and leave the rest (minus stopwords) for keyword processing [7]. For example, for the query “birth date of Abigail, the widow” the extraction ontology in Figure 5, with good recognizers, would discover that “birth date” references the *BirthDate* object set, that “Abigail” is a name in the *GivenName* object set, and that “widow” is a keyword. Hence, the query processing system would generate a query that joins over the relationship sets connecting the identified object sets in Figure 5, selects with the constraint *GivenName* = ‘Abigail’, and projects on the mentioned object sets—*Year* of *BirthDate* and *GivenName* for this query. The semantic index links to the pages on which (1) the name “Abigail” and a birth year are mentioned, and (2) the keyword “widow”

⁶ <http://code.google.com/p/duke/>

is present. Since the page in Figure 1 has both, the page-rank algorithm would place it high on its list.

- Advanced Form-based Query Processing. Because we process queries with extraction ontologies based on conceptual models, once an extraction ontology is identified as being applicable for a query, the system may use it to generate a form for advanced query processing. The query processing system treats all constraints in a free-form query conjunctively, but the generated form allows for the specification of negations and disjunctions as well [7].
- Cross-Language Query Processing. Since extraction ontologies are language independent, we can both semantically index and process queries in any language. (In our research we have implemented test cases for English, French, Japanese, and Korean.) We process cross-language queries by requiring that the extraction ontologies for each language have structurally identical conceptual-model instances. Thus, we are able to interpret a query with the extraction ontology in language L_1 and translate the query at the conceptual level to the extraction ontology in language L_2 . We can then execute the query over the semantic and keyword indexes to obtain a result in language L_2 , which can then be translated back into language L_1 for display to the user [3,7].
- *Conceptual-Model-Based Constraint Checking.* To address the Big Data issue of *veracity* in our family-history application, we envision applying the constraints declared in a conceptual model to check constraint violations. For example, a person should have only one mother. Because the data is obtained through information extraction and through other means such as crowd sourcing and wiki-like updates by the general public, we allow conflicting information to enter into the system, resulting in a myriad of constraint violations: “I’m my own grandpa”, as the saying goes, occurs in the actual (fairly massive) amount of data collected so far [8]. Big Data quality [9] will become a huge issue for our family-history application.
- *Conceptual-Model-Based Ontology Construction.* Ontology construction is one of the bottlenecks preventing the semantic web from achieving its envisioned potential as a Big Data application. Conceptual modeling can play a role in automating ontology construction. We have experimented with an approach to automated ontology construction, that takes a collection of tables all on some topic (e.g., *Statistics Canada*, <http://www.statcan.gc.ca/start-debut-eng.html>), interprets each table and reverse-engineers it into a conceptual-model instance, and integrates the conceptual-model instances into an ontology that covers the concepts and relationships discovered in the collection of tables [10].

The era of web-scale applications and Big Data is here to stay. As conceptual-modeling researchers we should look for ways to integrate our theory into the practice of Big Data. We see excellent opportunities in all four dimensions of Big Data (*volume*, *velocity*, *variety*, *veracity*) and expect that the community

can find more beyond those mentioned here in connection with our efforts to superimpose a web of knowledge over historical documents.

References

1. Zikopoulos, P.C., Eaton, C., de Roos, D., Deutsch, T., Lapis, G.: Understanding Big Data. McGraw-Hill, Inc., New York (2011)
2. Embley, D.W., Campbell, D.M., Jiang, Y.S., Liddle, S.W., Lonsdale, D.W., Ng, Y.-K., Smith, R.D.: Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering* 31(3), 227–251 (1999)
3. Embley, D.W., Liddle, S.W., Lonsdale, D.W., Tijerino, Y.: Multilingual ontologies for cross-language information extraction and semantic search. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 147–160. Springer, Heidelberg (2011)
4. Packer, T.L., Embley, D.W.: Cost effective ontology population with data from lists in ocred historical documents. In: Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing (HIP 2013), Washington, D.C., USA (to appear, August 2013)
5. Park, J.S., Embley, D.W.: Extracting and organizing facts of interest from ocred historical documents. In: Proceedings of the 13th Annual Family History Technology Workshop, Salt Lake City, Utah, USA (March 2013)
6. Embley, D.W., Zitzelberger, A.: Theoretical foundations for enabling a web of knowledge. In: Link, S., Prade, H. (eds.) FoIKS 2010. LNCS, vol. 5956, pp. 211–229. Springer, Heidelberg (2010)
7. Embley, D.W., Liddle, S.W., Lonsdale, D.W., Park, J.S., Shin, B.-J., Zitzelberger, A.J.: Cross-language hybrid keyword and semantic search. In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012 Main Conference 2012. LNCS, vol. 7532, pp. 190–203. Springer, Heidelberg (2012)
8. Cannaday, A.B.: Solving cycling pedigrees or “loops” by analyzing birth ranges and parent-child relationships. In: Proceedings of the 13th Annual Family History Technology Workshop, Salt Lake City, Utah, USA (March 2013)
9. Batini, C.: Data quality vs big data quality: Similarities and differences. In: Proceedings of the 1st International Workshop on Modeling for Data-Intensive Computing, Florence, Italy (October 2012)
10. Tijerino, Y.A., Embley, D.W., Lonsdale, D.W., Ding, Y., Nagy, G.: Toward ontology generation from tables. *World Wide Web: Internet and Web Information Systems* 8(3), 261–285 (2005)

What's Up in Business Intelligence?

A Contextual and Knowledge-Based Perspective

Marie-Aude Aufaure

Ecole Centrale Paris, MAS Laboratory, Chatenay-Malabry, France
Marie-Aude.Aufaure@ecp.fr

Abstract. The explosive growth in the amount of data poses challenges in analyzing large data sets and retrieving relevant information in real-time. This issue has dramatically increased the need for tools that effectively provide users with means of identifying and understanding relevant information. Business Intelligence (BI) promises the capability of collecting and analyzing internal and external data to generate knowledge and value, providing decision support at the strategic, tactical, and operational levels. Business Intelligence is now impacted by the Big Data phenomena and the evolution of society and users, and needs to take into account high-level semantics, reasoning about unstructured and structured data, and to provide a simplified access and better understanding of data. This paper will depict five years research of our academic chair in Business Intelligence from the data level to the user level, mainly focusing on the conceptual and knowledge level.

Keywords: Business Intelligence, Semantic Technologies, Content Analytics.

1 Introduction

Business Intelligence main objective is to transform data into knowledge for a better decision-making process. We have now entered the era of knowledge. Ubiquitous computing as well as the constant growth of data and information leads to new ways of interaction. Users manipulate unstructured data – documents, emails, social networks, contacts – as well as structured data. They also want more and more interactivity, flexibility, dynamicity. Users expect immediate feedback, and want to find information rather than merely look for it. Decision-making is more and more rapid and we need to automate more decisions. Moreover, the company tends to be organized in a collaborative way, called enterprise 2.0. All these evolutions induce challenging research topics for Business Intelligence, such as providing efficient mechanisms for a unified access and model to both structured and unstructured data. Extracting value from all these data, a crucial advantage in an hypercompetitive market, requires content analytics. In order to synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data, the use of data visualization techniques becomes critical. This leads to visual analytics, which combines automated analysis

techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets [1]. Business Intelligence is also impacted by big data, and need to take account of the heterogeneity and the volume of data sources as well as the need of response in real-time for extracting value from trusted data. Finally, Business Intelligence will integrate collaborative and social software. This is known as Business Intelligence 2.0 [2], which combines BI with elements from both Web 2.0 and the Semantic Web.

The organization of this paper is the following: Section 2 first overviews the evolution of Business Intelligence. Section 3 is dedicated to the presentation of research work we conducted in the academic chair in Business Intelligence during five years. Finally, Section 4 describes our future work and related challenges.

2 From Classical to Modern Business Intelligence

Business Intelligence (BI) refers to a set of tools and methods dedicated to collecting, representing and analyzing data to support decision-making in enterprises. BI is defined as the ability for an organization to take all input data and convert them into knowledge, ultimately, providing the right information to the right people at the right time via the right channel. During the two last decades, numerous tools have been designed to make available a huge amount of corporate data for non-expert users. Business Intelligence is a mature technology, widely adapted, but faces new challenges such as incorporating unstructured data into analytics. These challenges are induced by constantly growing amounts of available data. A key issue is the ability to analyze in real-time these data, taking their meaning into account. This amount of information generated and maintained by information systems and their users leads to the increasingly important concern of information overload. Personalized systems and user modeling [3] have thus emerged to help provide more relevant information and services to the user. The complexity of BI tools and their interface is a barrier for their adoption. Thus, information visualization and dynamic interaction techniques are needed for a better use of such tools. Semantic technologies [4][5] focus on the meaning of data and are capable of dealing with both unstructured and structured data. Having the meaning of data and a reasoning mechanism may assist a user during her analysis task.

Traditional BI systems can be extended with semantic technologies to capture the *meaning* of data and new ways of interacting with data, intuitive and *dynamic*. The vision of the CUBIST¹ project is to extend the ETL process to both structured and unstructured data, to store data in a triple store and to provide user-friendly visual analytics capabilities.

Fig. 1 (a) depicts the classical architecture of a Business Intelligence system in which data sources are structured and loaded in a data warehouse. Users can interact with restricted queries producing a static dashboard. In the CUBIST vision (Fig. 1 (b)), data sources are heterogeneous, both structured and unstructured and semantically stored in a triple store. Users can interact with flexible queries and dynamically perform visual analytics. Fig. 1 (c) depicts the most recent trend for BI taking into

¹ CUBIST EU FP7 project: <http://www.cubist-project.eu/index.php?id=378>

account *streams and semantics*. With the exponential growth of sensor networks, web logs, social networks and interconnected application components, large collections of data are continuously generated with high speed. These data are called "*data streams*": there is no limit on the total volume of data and there is no control over the order in which data arrive. These heterogeneous data streams [6] are produced in real time and consequently, should be processed on the fly. Then, they are maintained, interpreted and aggregated in the purpose of reusing their semantics and recommending relevant alerts to the targeted stakeholders in order to react to interesting phenomena occurring in the input streams. A precious decision-making value can be enhanced through the semantic analysis of data streams, especially while crossing them with other information sources. Real-time BI can be linked to Semantic BI, or can represent an independent evolution. In what follows, we consider that real-time BI integrates semantic technologies for being able to capture the meaning of large amounts of data.

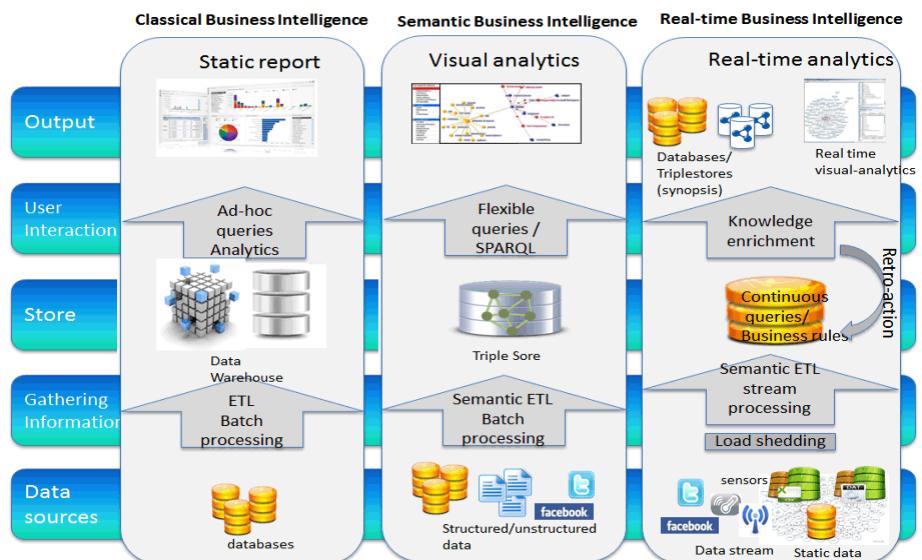


Fig. 1. Classical BI (a), Semantic BI (b), Real-Time BI (c)

In a Real-Time BI platform, multiple heterogeneous data sources can be connected, and data can be static or dynamic. The static data comes from standard databases or from open data, and does not change or in a minor way. Dynamic data comes as a stream, in a semantic format (RDF for example) or not (raw data). After their capture, data streams and static data are submitted to a set of semantic filters designed to achieve some specific business process. To manage infinite real-time data stream, the platform has to provide the ability to create persistent continuous queries, which allow users to receive new results when they become available.

Semantic filters are used to interconnect streams from various sources and to perform reasoning on these interconnected streams, possibly merged with static data (knowledge databases or ontologies). Semantic filters can also include summarizing

operators that extract subset of data in different form (uniform random samples, clusters, patterns, etc.). Load shedding [7] drops excess load by identifying and discarding the relatively less important data. The results of semantic filtering can be for example an alert, a new data stream, enrichment of ontology, feeding a dashboard, etc. Results can also cause feedback (stream connected back into the platform) in order to improve treatment or to add a context to the current treatments.

3 Semantic and Context for Business Intelligence

During five years, we have developed research and tools integrating structured and unstructured data for a more user-friendly decision-making process². The objective was to develop a knowledge layer allowing the end user to easily get the meaning of vast amounts of data, and visual analytics and querying tools for a user-friendly access to data. Fig. 2 summarizes the research work we developed:

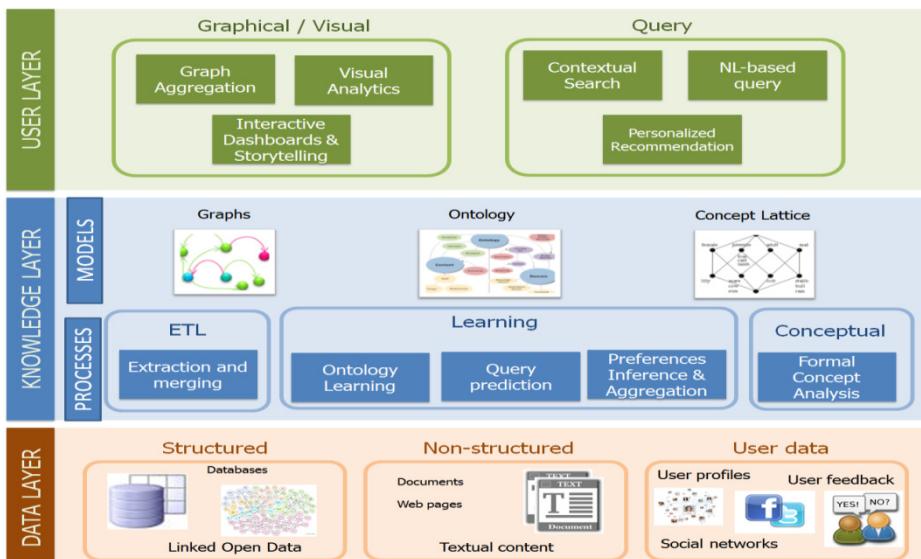


Fig. 2. A vision of Semantic and Contextual Business Intelligence

3.1 Data Layer and Models

The data layer is composed of a set of heterogeneous data sources: structured ones (databases and Linked Open Data), unstructured content (documents, web pages) and user data (user profiles and feedback, social networks). These data sources are structured by the mean of a knowledge layer. Graphs can be seen as a natural way for representing both structured and unstructured information. User's situation and interactions can be represented by a graph, and operators can be applied on a particular

² This work has been funded by SAP and EU FP7 projects CUBIST and PARLANCE

graph in response to business events. Moreover, one can benefit from graph operators and algorithms (shortest path, graph transformation) as well as social networks metrics (centrality). Semantic technologies, and particularly ontologies, can be useful for many applications. This model is central in our research work and used in many processes we defined. Formal Concept Analysis (FCA) captures hitherto undiscovered patterns in data. Formal concepts (also called concepts) are formalized as groups of objects associated with groups of attributes. Hierarchical relationships between these groups are formed and visualized by Galois Lattices in the form of a Hasse Diagram [8]. Information organized in this way has a close correlation to human perception but the challenging issue is to provide intuitive visualizations of Galois Lattices.

3.2 Knowledge Layer Processes

The first one is an *ETL process* (Extraction, Transformation, Load) extracting and merging graphs from relational databases, modeled using a complex graph structure (like typed attributed graphs). This transformation of relational databases into a graph allows the user to discover hidden relationships between objects [9]. However, this process still induces challenges mainly related to names resolution and scalability issues for merging graphs extracted from different databases.

The next set of processes is related to *learning*: ontology learning, query prediction based on users' OLAP sessions, and preferences inference and aggregation.

Ontology learning [10] is used to dynamically build a knowledge base, composed of ontology modules, from web abstracts and users' queries [11]. This process has been designed in an automatic and domain-independent way, exploiting unsupervised techniques and the web as a social scale learning source. We exploited these modular ontologies for semantic search, combining them with case-based reasoning. A case is defined by a set of similar queries associated with its relevant results. The case base is used both for ontology module learning and for contextualizing the search process. Module-based similarity is used to retrieve similar cases and to provide end users with alternative documents recommendations. Finally, with the rapid growth of structured data on the Web, referred as the Linked Open Data (LOD) including over 31 billion triples interlinked by around 504 million links, we have extended our approach and exploited DBpedia as a way to bootstrap the learning of linguistic patterns from unstructured data on the Web [12]. Subsequently the self-learned patterns are used for the extraction of new entities and ontology enrichment. In order to do this, we applied deep shallow syntactic analysis by using grammatical dependency analysis on Web snippets provided iteratively by a search engine according to automatically generated queries. Open research issues still remain. Among them, we can cite scalability issues and how to automatically find relevant data sources and ontologies for open domain or specific applications.

The *query prediction* process is related to infer the next possible OLAP query based on recent analytical sessions. In Business Intelligence systems, users interact with data warehouses by formulating OLAP queries aimed at exploring multidimensional data cubes. Being able to predict the most likely next queries would provide a

way to recommend interesting queries to users on the one hand, and could improve the efficiency of OLAP sessions on the other. In particular, query recommendation [13] would proactively guide users in data exploration and improve the quality of their interactive experience. Our framework for predicting the most likely next query and recommend it to the user relies on a probabilistic user behavior model built by analyzing previous OLAP sessions and exploiting a query similarity metric [14].

Preferences inference and aggregation are related to user model learning, critique-based mechanism for query refinement and sentiment analysis. The first two processes are related to spoken-dialogue in the context of the PARLANCE³ European project. Many current spoken dialogue systems for search are domain-specific and do not take into account the preferences and interests of the user. In order to provide a more personalized answer tailored to the user needs, we propose a spoken dialogue system where user interests are expressed as scores in modular ontologies, each ontology module corresponding to a search domain. This approach allows for a dynamic and evolving representation of user interests. Concepts and attributes in hierarchical ontology modules are associated with weight vectors expressing the interest or disinterest of a user on different levels of granularity. A key challenge for personalized mobile search is to tailor the system answers to the specific user and his current contextual situation. In particular, it is essential for recommender systems to perform preference adjustments based on user feedback, in order to modify the actual user behavior. However, regardless of the importance of user preference adjustments, it is not a trivial task. To tackle this challenge, we developed a preference-enabled querying mechanism for personalized mobile search by adjusting user preferences according to user's critiques and refining the queries with respect to the adjusted preferences [15].

Preferences aggregation deals with sentiment analysis [16]. Considering the impressive amount of unmediated opinions expressed by users in social network environments, we analyzed this data with the goal of automatically detecting the polarity of their opinions and perform recommendations. For this, we presented an approach to feature-level sentiment detection that integrates natural language processing with statistical techniques, in order to extract users' opinions about specific features of products and services from user-generated reviews [17].

The last process is related to **Formal Concept analysis** (FCA). We used FCA in the context of social network analysis [18], for classifying tweets on specific topics. We also use FCA to flexibly and efficiently build user communities. This entails a novel approach to represent dynamically evolving user preferences and interests. By analyzing the dialogue history of the user, interests are inferred and ontology modules for different domains are annotated with scores. The interests are used to perform formal concept analysis and to construct ad-hoc communities of users sharing similar interests, allowing a form of social search. By collaborative filtering we can share and recommend possibly interesting information and additional communities to users.

³ PARLANCE EU FP7 project:

<https://sites.google.com/site/parlanceprojectofficial/>

3.3 User Layer

The user layer is structured in two ways: by having graphical/visual support and by providing textual or formulation support.

Our research in ***information visualization*** has so far focused on 2 directions: (i) improving user experience when using BI dashboards and (ii) developing new visualization and interaction techniques for exploring data that are applicable to multiple domains (e.g. BI, intelligence analysis, social sciences). The main contributions from this work have been new ways to visualize data changes on charts, a new context aware annotation model for dashboards, as well as a set of domain independent interaction and visualization techniques for different data (usually in graph form).

A gap in the application of FCA to Business Intelligence concerns visual analytics. In FCA, the hierarchical relationships between concepts are traditionally displayed as a line diagram representation of the lattice. The concept lattice visualization can be greatly enhanced by visual analytics features and interlinked with best practices from known BI visualizations. A challenge is to manage and navigate the complex concept interrelationships, by condensing and clustering the results, and by sub-dividing and filtering data.

Graphs, and more specifically social networks, can have a huge size making them difficult to analyze and interpret. Producing meaningful summaries from complex graphs, taking into account multiple relations between nodes and various attributes in nodes, is a necessary step. We extended an aggregation algorithm, and defined two new aggregation criteria to improve the quality of the results and experimented them on various graphs.

As analysts continue to work with increasingly large data sets, data visualization has become an incredibly important asset both during sense making analysis, and when communicating findings to other analysts, decision makers or to a broader public. Individually and collectively, stories help us make sense of our past and reason about the future. Given the importance of storytelling in different steps of the analysis process it is clear there is a need to enhance visual analysis tools with storytelling support. We followed user-centered design approach to implement a storytelling prototype incorporated in an existing visual analysis dashboard.

Information access can also be managed through a ***textual or formulation support*** like NLP-based queries or for managing query reformulation and personalized access. A very promising use-case for Question-Answering (Q&A) over structured data is Business Intelligence. Understanding and converting an end-user's natural language input to a valid structured query in an ad-hoc fashion is still a challenging issue. Following this direction, we have proposed a framework for Q&A systems able to define a mapping between recognized semantics of a user's questions to a structured query model that can be executed on arbitrary data sources [19]. It is based on popular standards like RDF and SparQL and is therefore very easy to adapt to other domains or use cases. Personalization and recommendation techniques are useful to suggest data warehouse queries and help an analyst pursue its exploration. We defined a personalized query expansion component which suggests measures and dimensions to iteratively build consistent queries over a data warehouse [20]. Our approach leverages (a)

semantics defined in multi-dimensional domain models, (b) collaborative usage statistics derived from existing repositories of Business Intelligence documents like dashboards and reports and (c) preferences defined in a user profile.

4 Future Research Axis

Nowadays, in order to efficiently and effectively access the vast stream of information available on and off-line, the users and/or enterprises need to resort to flexible and efficient platforms or integration services.

The state of the art, however, presents various dilemmas between data, scale and temporary constraints; on the one hand, for example, common platforms and/or application which try to exploit information from available data stream (whatever they are) tend to still rely on standard keyword-based analysis and queries, and pure matching algorithms (using word frequencies, topics recentness, documents authority and/or thesauri) to find suitable information relevant to their needs. The result is that those platforms are not able to lead users into an intuitive exploration of large data streams because of their cumbersome presentations of the results (e.g. large lists of entries) and, above all, their missing interpretation of the data.

On the other hand, since data arrives continuously, fast one-pass algorithms are imperative for real-time processing of streams. Many solutions have been developed recently for processing data streams, however, none of them takes into account the semantic knowledge of data (stream reasoning). Moreover, the existing platforms do not provide any mechanism to retrieve other relevant information associated to those contents and, even if user feedback methods are proposed, it is really hard to detect the requested information because of inexperience and common lack of familiarity with the proposed visual tools.

Ontologies and, more generally, Linked Data can be now exploited due to its rich collection of structured information, as well as the “deep Web” of database-backed contents. The richness of these semantic data offers promising opportunities for identifying and disambiguating entities from one or multiple sources.

For this, the aim of a successful and effective platform which goal is to process and analyze dynamic stream of data should be the recognition and identification not only of those entries that are relevant to some high-level user need, but also the identification of the entities semantically related to this need and their disambiguation with respect to alternative similar entities.

Moreover, considering that one of the key aspects that has led to the popular success of user-generated content is the possibility to express (and read) unmediated individual opinions, we believe that a successful platform should try to integrate this knowledge for improving its performance. In fact, nowadays, users are currently expressing their opinions and interests on a wide range of entities, products and services. The importance of this data in search and decision making processes is therefore evident. In fact, due to the ever-growing amount of different entities and/or product with similar characteristics, the users are not only aiming at analyzing and searching the best entities that match their interests but their goal is also to find those that agree

with the vast of majority of the users who already have an opinion about the considered entity. In other words, they are often searching for authentic, user-generated reviews to orient their search and decisions.

The platform we aim to build will provide an environment for real users and/or enterprises to analyze, search, detect and visualize real world entities within a dynamic stream of massive data and organize the list of relevant results based on knowledge and contextual information.

Acknowledgments. I would like to acknowledge all PhD students, postdoctoral researchers and internships having worked in the team during the past five years.

References

1. Keim, D.A., Kohlhammer, J., Ellis, G., Mansmann, F. (eds.): Mastering the Information Age: Solving Problems with Visual Analytics, Thomas Müntzer (2010)
2. Trujillo, J., Maté, A.: Business Intelligence 2.0: A General Overview. In: Aufaure, M.-A., Zimányi, E. (eds.) eBISS 2011. LNBIP, vol. 96, pp. 98–116. Springer, Heidelberg (2012)
3. Kobsa, A.: Generic user modeling systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 136–154. Springer, Heidelberg (2007)
4. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
5. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
6. Aggarwal, C. (ed.): Data Streams. Models and Algorithms. Advances in Database Systems, vol. 31. Springer (2007)
7. Tatbul, N., Cetintemel, U., Zdonik, S.: Staying FIT: Efficient Load Shedding Techniques for Distributed Stream Processing. In: International Conference on Very Large Data Bases (VLDB 2007), Vienna, Austria (2007)
8. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations Edition. Springer (1999)
9. Soussi, R., Cuvelier, E., Aufaure, M.A., Louati, A., Lechevallier, Y.: DB2SNA: an All-in-one Tool for Extraction and Aggregation of underlying Social Networks from Relational Databases. In: Ozyer, T., et al. (eds.) The Influence of Technology on Social Network Analysis and Mining, Springer (2012) ISBN 978-3-7091-1345-5
10. Buitelaar, P., Cimiano, P. (ed.): Ontology Learning and Population: Bridging the Gap between Text and Knowledge. Series Information for Frontiers in Artificial Intelligence and Applications. IOS Press (2008)
11. Ben Mustapha, N., Aufaure, M.A., Baazaoui-Zghal, H., Ben Ghezala, H.: Query-driven approach of contextual ontology module learning using web snippets. Journal of Intelligent Information Systems (2013)
12. Tiddi, I., Mustapha, N.B., Vanrompay, Y., Aufaure, M.-A.: Ontology Learning from Open Linked Data and Web Snippets. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM-WS 2012. LNCS, vol. 7567, pp. 434–443. Springer, Heidelberg (2012)
13. Giacometti, A., Marcel, P., Negre, E.: A framework for recommending OLAP queries. In: Proc. DOLAP, Napa Valley, USA, pp. 73–80 (2008)
14. Aufaure, M.-A., Kuchmann-Beauger, N., Marcel, P., Rizzi, S., Vanrompay, Y.: Predicting your next OLAP query based on recent analytical sessions. In: Bellatreche, L., Mohania, M.K. (eds.) DaWaK 2013. LNCS, vol. 8057, pp. 134–145. Springer, Heidelberg (2013)

15. Hu, B., Vanrompay, Y., Auaure, M.-A.: PQMPMS: A Preference-enabled Querying Mechanism for Personalized Mobile Search. In: Faber, W., Lembo, D. (eds.) RR 2013. LNCS, vol. 7994, pp. 235–240. Springer, Heidelberg (2013)
16. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
17. Cataldi, M., Ballatore, A., Tiddi, I., Auaure, M.A.: Good Location, Terrible Food: Detecting Feature Sentiment in User-Generated Reviews. International Journal of Social Network Analysis and Mining (2013)
18. Wasserman, S., Faust, K.: Social Network Analysis: Methods and applications. Cambridge University Press (1994)
19. Kuchmann-Beauger, N., Brauer, F., Auaure, M.A.: QUASL: A Framework for Question Answering and its Application to Business Intelligence. In: Seventh IEEE International Conference on Research Challenges in Information Science (2013)
20. Thollot, R., Kuchmann-Beauger, N., Auaure, M.-A.: Semantics and Usage Statistics for Multi-Dimensional Query Expansion. In: Lee, S.-g., Peng, Z., Zhou, X., Moon, Y.-S., Ulland, R., Yoo, J. (eds.) DASFAA 2012, Part II. LNCS, vol. 7239, pp. 250–260. Springer, Heidelberg (2012)

Modeling and Reasoning with Decision-Theoretic Goals

Sotirios Liaskos¹, Shakil M. Khan², Mikhail Soutchanski³, and John Mylopoulos⁴

¹ School of Information Technology, York University, Toronto, Canada
liaskos@yorku.ca

² Department of Computer Science and Engineering, York University, Toronto, Canada
skhan@cse.yorku.ca

³ Department of Computer Science, Ryerson University, Toronto, Canada
mes@cs.ryerson.ca

⁴ Department of Information Engineering and Computer Science, University of Trento, Italy
jm@disi.unitn.it

Abstract. Goal models have found important applications in Requirements Engineering as models that relate stakeholder requirements with system or human tasks needed to fulfill them. Often, such task specifications constitute rather idealized plans for requirements fulfillment, where task execution always succeeds. In reality, however, there is always uncertainty as to whether a specification can/will actually be executed as planned. In this paper, we introduce the concept of decision-theoretic goals in order to represent and reason about both uncertainty and preferential utility in goal models. Thus, goal models are extended to express probabilistic effects of actions and also capture the utility of each effect with respect to stakeholder priorities. Further, using a state-of-the-art reasoning tool, analysts can find optimal courses of actions/plans for fulfilling stakeholder goals while investigating the risks of those plans. The technique is applied in a real-world meeting scheduling problem, as well as the London Ambulance Service case study.

Keywords: Information Systems Engineering, Goal Modeling, DT-Golog, Decision Theory.

1 Introduction

Goal-Oriented Requirements Engineering is founded on the premise that functional requirements for information systems can be derived from stakeholder goals through a systematic process [1]. For example, the goal *Schedule a Meeting* in a university setting might be fulfilled by a system that supports a set of functions (gather constraints automatically, find free slots, send out reminders, etc.) as well as actions carried out by external actors (participants, a meeting initiator etc.). When the designer selects an alternative for fulfilling top-level stakeholder goals and generates a design, the implicit claim is that the design will fulfill every instance of the goal (e.g., successfully schedule and hold every requested meeting).

Unfortunately, the world is not that simple. The design – which implements a generic plan for fulfilling the goal – may actually fail for a number of reasons, including limited resources, bad scheduling, unexpected obstacles, and more. For instance, participants

may provide inaccurate constraints or maintain incomplete on-line calendars. Or, the email with the meeting invitation may include the wrong time or room. Even sending the email per se often does not guarantee its receipt, especially when mail servers or anti-spam filters are not appropriately maintained. Hence, actions – carried out by the system or actors in its environment – produce effects that vary in uncontrollable ways. In other words, a design may fail to fulfill instances of a goal due to violation of implicit domain assumptions and axioms [1], such as those pertaining to the expected effects of system or user actions.

To address such uncertainties, stakeholders may want to posit probabilistic requirements, such as *Meeting scheduling requests will be fulfilled 95% of the time* [2]. Given such requirements, it is the task of the designer to come up with a plan that will succeed within the probabilistic constraints of the requirement. At the same time, however, stakeholders wish to maintain the multi-objective nature of alternatives analysis. Thus, potentially conflicting goals such as *Quick Scheduling vs. Maximize Attendance* or *Keep Secretary Unburdened vs. Quality of Schedule* may each be served better by different designs. Stakeholders may be willing to exchange an increased probability of failure with an increased value in one or more of those objectives in case of success. In these circumstances, searching for a suitable design is a process of finding designs that offer the best combination of quality, based on stakeholder preferences, and likelihood of success, i.e. the best expected value.

In this paper, we introduce the concept of *decision-theoretic goals*, which combines the merits of their probabilistic [2] and their preferential [3] cousins in order to capture both probabilistic uncertainty and preferential utility in goal models. To achieve this, we extend the preference and priority-enabled goal modeling language we proposed in [3] to allow representation of probabilistic actions, that is, actions that do not have just one unique effect but a probability distribution over possible effects/outcomes. Utility functions, on the other hand, assign different desirability measures to different such action outcomes. The extended goal model is then translated into *DT-Golog*, a formal specification language that combines ideas from dynamic domain specification languages and Markov Decision Processes (MDPs) [4,5]. A DT-Golog reasoning tool is then used to evaluate alternative designs by which the specified goals are fulfilled with optimal expected value. This way, both likelihood and value are considered when searching for good solutions to the given requirements problem.

We organize the paper as follows. In Section 2 we present our goal modeling notation and in Section 3 we show how we extend it to allow for decision-theoretic analysis. In Section 4 we show what kinds of automated reasoning the technique enables. Then, in Section 5 we report on an application to a real-world meeting scheduling problem as well as the London Ambulance Service (LAS) case and discuss tool performance. Finally, we survey related work in Section 6 and conclude in Section 7.

2 Goal Models

Goal models ([1,6]) have been found to be effective in concisely capturing large numbers of alternative sets of low-level tasks, operations, and configurations that can fulfill high-level stakeholder goals. In Figure 1, a (simplified) goal model for scheduling meetings is depicted. The model shows how the high-level goal of a meeting organizer to

Have a Meeting Scheduled is analyzed into the particular subgoals and actions that are needed for the goal to be attained. The model primarily consists of *goals* (also: *hard-goals*) and *tasks*. Goals – the ovals in the diagram – are generally defined as states of affairs or conditions that one or more actors of interest would like to achieve [6]. Tasks, on the other hand, – the hexagonal shapes – describe particular activities that the actors perform in order to fulfill their goals.

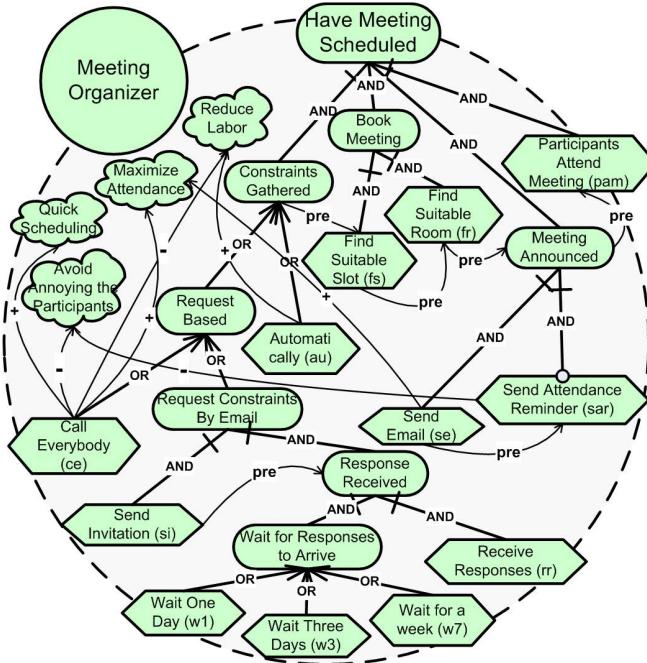


Fig. 1. A goal model

Goals and tasks are connected with each other via AND- and OR-decompositions. By AND-decomposing a goal into other subgoals or tasks, we indicate that the satisfaction of each of its children is necessary for the decomposed goal to be fulfilled. However, children of AND-decompositions can be designated as *optional* through a circular annotation added on their top, such as *Send Attendance Reminder* in the figure. On the other hand, if the goal is OR-decomposed into other goals or tasks, then the satisfaction of one of these goals or tasks suffices for the satisfaction of the parent goal.

The *order* in which goals and tasks are satisfied and performed respectively is relevant. To express constraints over satisfaction ordering we use a *precedence link* ($\xrightarrow{\text{pre}}$). A precedence link drawn from a goal/task to another goal/task, indicates that satisfaction/performance of the target of the link cannot begin unless the origin is satisfied or performed. Thus, the precedence link from *Find Suitable Room* to *Meeting Announced* indicates that unless the former is performed, none of the tasks below the latter can be performed. Furthermore, the *negative precedence link* ($\xrightarrow{\text{npr}}$) indicates that performance of the link target cannot start if the element at the origin of the link has been satisfied.

Moreover, *soft-goals* (the cloud-shaped elements) represent goals whose fulfillment does not have a clear-cut satisfaction criterion. Since satisfaction of soft-goals cannot be established in a crisp manner, the degree by which they are satisfied is assessed through evidence of satisfaction of other goals. In the goal model, this is represented through positive *helps* (\rightarrow^+) and negative *hurts* (\rightarrow^-) *contribution links* drawn from goals and tasks to soft-goals.

The AND/OR decomposition implies a number of sequences of leaf-level tasks that can satisfy the top level hard goal. We call such sequences *plans*. The variability of such plans emerges both due to the existence of OR-decompositions and optional sub-goals in the AND/OR tree, allowing for different subsets of tasks that can fulfill the root goal, and due to the fact that a given subset (i.e., a solution to the AND/OR tree) can be ordered in different ways subject to $\xrightarrow{\text{pre}}$ and $\xrightarrow{\text{npr}}$ constraints. Furthermore each plan has a different impact to high-level soft-goals. Back in Figure 1, a plan that includes calling everybody to acquire constraints has a negative impact to the soft-goal *Reduce Labour* and should be avoided if that soft-goal is important. It would be a good plan, however, if *Quick Scheduling* were a high priority goal.

3 Goals, Probabilities and Utilities

3.1 Decision-Theoretic Goals

In the standard notation we described above, performance of tasks is assumed to bring about the desired result with certainty. In reality, however, tasks have multiple intended or unintended outcomes, each with different likelihood. As such, task performance does not guarantee goal achievement. To model and reason about this uncertainty, the traditional concept of a goal has been extended to include a probability of success to it [2,7]. Hence, probabilistic goals describe a desired state of affairs as well as a minimum probability value for this state to be successfully reached.

To this, however, we wish to add here another dimension: that of *utility* as measured by the impact that solutions of the goal have to high-level qualities and stakeholder priorities thereof. Thus, *decision-theoretic goals* require maximization of *expected utility*, which combines probability of success and utility. Thus:

“Have Meeting Scheduled”, optimally

requires that a meeting is scheduled while maximizing expected utility. Nevertheless, since expected utility combines probability and utility, it is possible that the plan with the optimal expected utility score has a forbiddingly low success probability. Thus goal:

“Have Meeting Scheduled”, optimally, prob 0.7

demands that we wish to schedule the meeting optimally but also ensure that the probability of success of the optimal plan exceeds 0.7. Hence, decision-theoretic goals prescribe both the quality that plans to achieve them must meet, in terms of satisfying high-level preferences, and our risk tolerance with respect to those plans.

To allow reasoning about decision-theoretic goals we extend the standard goal modeling formalism with the following elements: *domain predicates*, which model the state features of the domain, *effect tables*, which model possible effects of tasks and their probabilities, *attainment functions* and *utility tables*, which connect the state of the

Table 1. Effect, Utility, and Priority Tables

Task: Receive Responses			
Affects (decision variables): <i>excellentResponsesReceived</i> , <i>adequateResponsesReceived</i> , <i>tooFewResponsesReceived</i>			
Depends on (condition variables): <i>waited1Day</i> , <i>waited3Days</i> , <i>waited1Week</i>			
Domain Predicates		p	
1.	<i>waited1Day</i>	<i>excellentResponsesReceived</i>	0.0
2.		<i>adequateResponsesReceived</i>	0.33
3.		<i>tooFewResponsesReceived</i>	0.67
4.	<i>waited3Days</i>	<i>excellentResponsesReceived</i>	0.17
5.		<i>adequateResponsesReceived</i>	0.5
6.		<i>tooFewResponsesReceived</i>	0.33
7.	<i>waited1Week</i>	<i>excellentResponsesReceived</i>	0.83
8.		<i>adequateResponsesReceived</i>	0.17
9.		<i>tooFewResponsesReceived</i>	0.0
Attainment Formula: (<i>adequateResponsesReceived</i> \vee <i>excellentResponsesReceived</i>)			

(a) An Effect Table

Softgoal: Avoid Annoying the Participants		
Depends on: <i>calledEverybody</i> , <i>reminderArrived</i>		
Domain Predicates		u
<i>calledEverybody</i>	<i>reminderArrived</i>	0.0
	<i>not reminderArrived</i>	0.3
<i>not calledEverybody</i>	<i>reminderArrived</i>	0.7
	<i>not reminderArrived</i>	1.0

(b) A Utility Table

<i>Avoid Annoying the Participants</i>		0.1
<i>Quick Scheduling</i>		0.2
<i>Reduce Labor</i>		0.7

(c) A Priority Table

domain with the achievement of goals and soft-goals, respectively, and *priority tables*, whereby we prioritize goals. We discuss each of these extensions below.

3.2 Representing State and Probabilistic Effects

The first step in our extension is to explicate what is true in the environment before, while and after the tasks of the goal model are performed. To do so we use *domain predicates*. Domain predicates represent fixed facts about the domain – e.g. *available(secretary)* or *has(projector, meetingRoom)* – or facts that may vary due to the performance of tasks or exogenous reasons – e.g. *invitationsSent* or *requested(meetingRoom)*. Each combination of truth values of the domain predicates determines the state in which the process for fulfilling the goals is; let S denote the set of all such combinations. Note that we do not model states explicitly in our proposal; rather we only model state features representing individual changeable properties domain.

Let us now focus on tasks. Ideally, performance of a task by an agent implies that certain facts in the domain change in a deterministic way, leading the system to a new state with certainty. In reality, as we claimed, this cannot always be assumed. Firstly, the outcomes of some tasks rely on chance due to their nature. For example the task *Find Suitable Time Slot* may or may not lead to a situation where *slotFound* holds, depending on the scheduling constraints at hand. Secondly, there are tasks that have one expected and/or desired outcome, but they always run a probability of failure. Thus, the task *Send Invitations* will most probably lead to the fact *invitationsReceived* being true, but there is always a probability of the same fact being false due to a number of factors, such as infrastructure error (server down, anti-spam false positives) or human error (accidental deletion or mishandling of email). Human actors in particular may have their own unidentified and conflicting goals that prevent them from acting as prescribed. Thus, we are interested in representing *probabilistic effects* of tasks – that is effects that, for a variety of reasons, may lead to different outcomes with different likelihood.

To do this we first associate each task with a number of effects that can potentially be brought about upon the task’s performance. Effects are represented using domain predicates. Thus, performance of the task *Receive Responses* may or may not have an

effect that *adequateResponsesReceived*, meaning that important participants responded but not many others, versus the competing (mutually-exclusive in our case) effects *tooFewResponsesReceived*, meaning that too few or none of the important participants responded and *excellentResponsesReceived*, meaning that a very satisfactory amount of responses has arrived, including the important participants. Each of those effects occurs with a certain probability given different conditions.

We can represent these probabilities using a decision table such as that of Table 1a; we call it the *effect table* for the task. The table actually represents the probability distribution over possible effects of the task *Receive Responses*. It contains one or more *decision variables*, which represent possible configurations of effects that the task can bring about, as well as one or more *condition variables*, which are the variables which the probabilities of various value configurations of the decision variables depend on. Both decision and condition variables are drawn from the set of domain predicates. Each combination of condition and effect configurations occurs with a certain probability.

In the example of Table 1a, there are three decision variables (domain predicates: *excellentResponsesReceived*, *adequateResponsesReceived* and *tooFewResponsesReceived*) and the probability that a certain combination of truth values occurs depends on three condition variables that have to do with how long the initiator waited after the original invitation (domain predicates: *waited1Day*, *waited3Days* and *waited1Week*). Thus the probability that adequate responses will arrive within the first three days is 0.5. Note that, in the particular example, both decision and condition variables are mutually exclusive. In the general case, arbitrary combinations of values can be considered.

3.3 Redefining Goal Satisfaction

Hard-goals and Probabilistic Effects. The probabilistic interpretation of task effects necessitates certain refinements to the goal model of Figure 1. Firstly, satisfaction of hard-goals does not exactly reflect the AND/OR structure of the underlying subtree anymore, because it is now measured by the effect of the underlying tasks and not by the mere fact that the tasks are performed. Hence, we define satisfaction of the goal based on the desirable effects of the task.

In the case of multiple effects, as in *Receive Responses* of Table 1a, we construct the *attainment formula* of each task and hard-goal exclusively based on domain predicates, which signifies what effects must be brought about to consider a goal or task satisfied or performed. In our example, the attainment formula of task *Receive Responses* could be *excellentResponsesReceived* \vee *adequateResponsesReceived*. Satisfaction of higher level hard-goals is defined via conjunctions or disjunctions of attainment formulae of tasks depending on the corresponding AND/OR structure. Thus, the attainment formula of *Book Meeting* is *slotFound* \wedge *roomBooked*, each being, in turn, predicates describing probabilistic effects of tasks *Find Suitable Slot* and *Find Suitable Room*, respectively. Note that it is in the discretion of the modeller to redefine satisfaction conditions, by, for example, setting the attainment formula of *Receive Responses* to be just *excellentResponsesReceived* – hence stricter than the previous one.

Assessing Soft-goal Satisfaction. As with hard-goals, in light of probabilistic effects of tasks, a refined model of the satisfaction of soft-goals should also depend on the actual outcome of task performance, rather than the mere fact that a task was performed.

For example, the claim that the task *Send Attendance Reminder* contributes negatively to the soft-goal *Avoid Annoying the Participants* can be supported only if the reminder actually went through – if not, no annoyance can reasonably be assumed. Thus, contribution links are refined into relationships between domain predicates and soft-goals. More specifically, similarly to the attainment formula we saw above, each soft-goal g of the goal graph is assigned an *attainment function* u_g that maps the set S of all possible truth assignments of domain predicates to an interval of real numbers: $u_g : S \mapsto [0, 1]$. Thus, different configurations of truth values for the domain predicates imply a potentially different value for the attainment function of the soft-goal at hand. The higher the attainment value, the more the soft-goal is believed to be satisfied. In the interval $[0, 1]$, 1.0 represents full satisfaction of the soft-goal and 0.0 its full denial. Thus, in effect, we quantify the originally qualitative contributions of the goal model – we suggest how below.

We found that representation of attainment functions is also possible using a tabular format such as that of Table 1b – we call it the *utility table*. The variables used in the table represent the domain predicates that influence the satisfaction of the soft-goal. Each combination of truth values of those domain predicates is associated with the actual attainment value (seen as a utility value) of the soft-goal at hand. Thus, attainment values express utility (with respect to the soft-goal at hand) of the situation(s) that is/are described by each truth value combination. In Table 1b, a possible attainment function for the soft-goal *Avoid Annoying the Participants* is shown. Attainment of that goal largely depends on whether the meeting organizer has called all participants on the phone to gather constraints, expressed through domain predicate *calledEverybody* as well as whether s/he has (successfully) sent them reminders to attend the meeting, modeled through the domain predicate *reminderArrived* – other predicates are irrelevant. In the utility table, different combinations of truth values of these domain predicates imply a different attainment value for the soft-goal, shown in the last column. Thus, according to the table, in any state $s \in S$ in which predicate *calledEverybody* is true and predicate *reminderArrived* is false, the attainment value for goal *Avoid Annoying the Participants* (for short: *AvoidAnP*) is $u_{\text{AvoidAnP}}(s) = 0.3$.

Aggregating Utilities Through Priority Profiles. Utility tables show how each state of the domain implies a different attainment value for a particular soft-goal. To assign to each state a universal “goodness” value which combines all soft-goals of interest we use *priority tables* [3]. A priority table is a representation of the relative importance of soft-goals, in form of a weighted numeric combination. They can include any subset of soft-goals from the goal model. Table 1c shows a priority table with three soft-goals. In the case of an hierarchical organization of soft-goals, we can elicit priority tables for each decomposition of the hierarchy, combine them in a larger profile containing only leaf-level soft-goals and continue our analysis with those (cf. [8]).

Given a priority profile and its weights we can construct a linear combination of attainment formulae of individual soft-goals expressing a measure of global utility U , which we call *total utility*. This measure is used for optimization as we describe below. More formally, let w_1, w_2, \dots, w_i be the priority values for soft-goals of interest g_1, g_2, \dots, g_i . Then, the total utility U for the goal model in state $s \in S$ is $U(s) = \sum_i w_i \times u_{g_i}(s)$. Thus, as per Table 1c, 0.1, 0.2 and 0.7 are the priority values

for soft-goals *Avoid Annoying the Participants* (for short: *AvoidAnP*), *Quick Scheduling* and *Reduce Labor* respectively. In a state s where soft-goal *Avoid Annoying the Participants* is satisfied by, e.g. 0.3 and *Quick Scheduling* and *Reduce Labor* are satisfied by 0.9 and 0.5 respectively, the total utility value U for that state will be:

$$\begin{aligned} 0.1 \times u_{\text{AvoidAnP}}(s) + 0.2 \times u_{\text{QuickScheduling}}(s) + 0.7 \times u_{\text{ReduceLabor}}(s) \\ = 0.1 \times 0.3 + 0.2 \times 0.9 + 0.7 \times 0.5 = \mathbf{0.56} \end{aligned}$$

Getting the Numbers. The quantitative measures we discuss above occur both in the form of probabilities and in the form of utility/priority values. Overall, while we focus in this paper on the technical representation and reasoning aspects, we believe that there are solid methods and experience in terms of eliciting probabilities and utility measures [9]. As we demonstrate below, probability numbers can come from either simple measurements in the domain or, in the absence of such, subjective judgement by the modellers. Further, there is a variety of ways by which utility and priority numbers can be found, including prominent requirements prioritization techniques such as AHP [10,8]; both Tables 1b and 1c can be results of AHP’s pairwise comparisons. Even subjective ad-hoc assessment is a realistic possibility: it has been found that even if the numbers are not exact, they may be good enough to make correct informed decisions [11]. Otherwise, numerical attainment values are expressions of utility and as such can be obtained through more systematic techniques such as reward elicitation [12].

4 Reasoning about Decision-Theoretic Goals

4.1 Integrating Decision-Theoretic Planning

The above extensions are useful for performing automated reasoning about goal satisfaction under probabilistic effects, utilities and soft-goal priorities. To enable this, the extended goal model is translated into *DT-Golog* [4,5]. DT-Golog is a formal language for modeling and reasoning about dynamic domains under uncertainty, through combining logical and procedural action theory specification and Markov Decision Processes (MDPs). A DT-Golog specification consists of constructs that represent state features, called *fluents*, as well as agent *actions* that bring the world from one *situation*, where some fluents are true, to another, where the same fluents or different ones might be true. In the action theory specification, DT-Golog programs “glue” actions together in a procedural manner in order to describe ways to achieve goals. Moreover, precondition and successor state axioms define what needs to be true in order for an action to be performed and how exactly fluents are affected by each action, respectively. The effects of actions are multiple and each with a different probability. Further, both actions and fluents are used to define utility functions. This way, DT-Golog can search for *policies* (i.e., nested branching statements prescribing what action to take depending on a condition) within constraints imposed by the program. A returned policy maximizes the *total accumulated expected utility* defined as the (gradually discounted) sum of the products of total utility values and the probability that each such value is obtained when following a remaining portion of the policy. In addition to a policy, DT-Golog also returns a probability of successful termination that sums probabilities of all branches in which a given program runs to completion.

The generation of a DT-Golog specification from the extended goal model in order to allow such reasoning, is based on translating tasks into actions, and domain predicates into fluents. Further, precedence links inform the formation of precondition axioms and effect tables translate into successor state axioms and probabilistic effects. DT-Golog procedures mimic the hard-goal structure, while the soft-goal structure is translated into a utility function. For the interest of space, the formal translation details appear in our longer technical report.

4.2 Querying for Optimal Solutions

Let us return to the example of Figure 1 and discuss different kinds of decision-theoretic goals we can reason about using DT-Golog with the generated specification.

Optimizing Expected Utility. Decision-theoretic goals of the form “*Schedule Meeting*”, *optimally* are satisfied by a policy p of the translated goal model G , iff p brings about the maximum accumulated expected utility in G . The necessary probability and utility measures are drawn from the appropriately translated effect, utility and priority tables we saw above. Thus, in Figure 1 and assuming we have introduced effect and utility tables for each of the involved tasks and soft-goals accordingly (which we do not present due to space constraints), by setting all soft-goals to be of equal priority we find a policy with total accumulated expected utility 2.7. The policy includes success plans (i.e. branches of the policy in which all tasks are successfully performed) such as $[si, w7, rr, fs, fr, se, sar, pam]$ (referring to abbreviations in the parentheses inside the task symbols). DT-Golog informs us also that the total probability of successful termination of the policy is 0.4.

The result is, of course, sensitive to probability and utility values. Thus, if we assume that soft-goals follow the priority values of Table 1c, instead of having equal priority as we assumed above, the resulting policy, with accumulated expected utility 3.1 and probability of success 0.34, includes success plans such as $[au, fs, fr, se, sar, pam]$. Clearly, the increased importance of soft-goal *Reduce Labour* in the priority table favours the choice of automated constraint gathering *au*.

Testing Probability Thresholds. The other kind of decision-theoretic goals that we saw has the form “*Schedule Meeting*”, *optimally, prob c*, where c is a probability value. Such a decision-theoretic goal is satisfied by a policy p of the translated goal model G iff p has the maximum accumulated expected utility in G and p has a probability of success greater or equal to c . Thus, DT-Golog simply tests if the optimal policy has a probability of success above c . For example, the second of the above optimal policy has a success probability of 0.34, meaning that, if we also had a probability threshold c of, say, 0.7, DT-Golog would report failure to find suitable policy. Note that optimality is defined in a global sense and independent of the probability threshold.

5 In Practice

5.1 A Meeting Scheduling Study

As a preliminary test of the feasibility of our modeling technique, we applied it to a meeting scheduling problem that occurs in our workplace. Our *SE@York* seminars are

events that we organize at York University and feature regular talks by visiting or resident software engineering scholars and PhD students. The first author is the meeting initiator of the SE@York meetings and has access to relevant data sources. Potential participants are professors and graduate students of the IT and CS departments. The standard request-based constraint acquisition method is performed by the initiator as seen in the model of Figure 1. In terms of quality goals, the real concern of the organizers is to have good attendance. To a lesser extend they would like to have the meeting scheduled as quickly as possible, for varying reasons including that e.g. a visitor speaker is leaving the country or running out of patience.

Getting the Numbers. In our domain, probabilistic data comes from the initiator's email archives (constraint requests and responses) as well as the paper-based room booking logs. The numbers presented in the effect table of task *Receive Responses* in Table 1a are actual values coming out of our data. The email archive data also allow us to calculate the probability that a slot will eventually be found (0.83, in our case). The room booking logs, on the other hand, allow us to calculate the probabilities that the meeting room will be available. For our study, we simply looked at the probability that the room is available at any workday from 9am to 5pm in January. A successful plan for having a meeting properly scheduled is defined to be one in which at least half of the responses have arrived prior to deciding on a time slot (so this is our semantics of the effect *adequateResponses*) and a time slot as well as a meeting room is found immediately. These success conditions are defined accordingly through attainment formulae.

To elicit utilities we make use of the Analytic Hierarchy Process (AHP) [8], focussing on soft-goals *Maximize Attendance* and *Quick Scheduling*. Thus we set $U(s) = 0.75 \times u_{\text{MaximizeAttendance}}(s) + 0.25 \times u_{\text{QuickScheduling}}(s)$. These two attainment formulae are defined also through pair-wise comparisons on the three probabilistic effects of the task *Receive Responses* as well as on the effects representing the three children of the goal *Wait for Responses to Arrive*.

Reasoning. We focus on the problem of how long the organizer must wait before deciding a slot. For the above utilities, which represent our actual preferences, the optimal solution is to wait for seven days. The probability of success in that case is 0.5. Should *Quick Scheduling* be more important than *Maximize Attendance* – and in our SE@York meetings there have been such cases – after swapping the priority weights, the optimal solution is to try waiting for 1 day. This is due to the fact that waiting less (e.g. *waited1Day*) has much higher utility now. But this solution has lower probability of success, 0.17, since within 1 day adequate responses may have not been received, leading to higher failure probability of the task *Receive Responses*. To see why these probability numbers appear to be low compared to our intuition one must remember that we define ourselves through attainment formulae what constitutes a successful plan.

5.2 Adding Detail

Our use of DT-Golog with the specification that is generated from the semi-formal goal model, exploits only a subset of DT-Golog's expressive power. To further study how DT-Golog's expressive capabilities are applicable to the requirements analysis problem, an application to the well known London Ambulance Service (LAS) [13] case was also performed. The application is described in detail elsewhere [14] – here we focus on key

features. The particular case concerns the problem of managing a fleet of ambulances to respond to emergency incidents in the city of London, UK. What makes the case particularly interesting for our purposes is the explicit performance requirements that can be imposed in the form of an exact probability distribution of allowable ambulance response times. More specifically, concrete performance and reliability requirements can be set for candidate dispatch strategies. Thus, we can demand that a request is responded to within 14 minutes of the time a call is placed, that activation time (call receipt and decision) should always be made in less than 3 minutes, or that travel time to the incident should be 11 minutes 95% of the time and 8 minutes 50% of the time.

To search for designs that meet these performance objectives, extension of the initial DT-Golog specification needs to be performed by adding detail in a number of ways. Firstly, domain information is added in the form of particular instances of objects, agents and contexts that are involved in the LAS operations. Thus, the geography of three city regions is modeled using 10x10 grids. Each hospital, ambulance, incident etc. is represented as a DT-Golog fact and occupies at a given point in time a particular cell in the grid, representing its geographical position. Actions and fluents are relativised to particular objects through parameters. Thus, a fluent of the type $carLocation(c,l,t,s)$ is used to represent that an ambulance c is at location l at time t in situation s – the location is represented through a term $loc(x, y)$, where x and y are co-ordinates in the grid. Actions also have a temporal argument, with which their duration is encoded.

To allow analysis of different dispatch strategies, each expected to have different performance characteristics, Golog procedures describing those strategies are written. These are more complex than translations of AND/OR structures that the framework we described above produces. Furthermore, the utility functions are an essential part of each strategy, as they describe the chosen optimization approach. Thus, aspects such as the familiarity of an ambulance driver in an area or the effect of personnel fatigue are modeled through appropriately structured utility tables.

Moreover, *simulation* is necessary when there is a need to model random variables representing exogenous events. In the LAS case, these are the occurrence of emergency incidents. A Poisson distribution of incidents is assumed with various arrival frequency scenarios. Different dispatch strategies are then repeatedly tried for a large number of requests. The response times are counted/averaged and compared with the set requirements, allowing better understanding of the behaviour of different response strategies.

5.3 Tool Performance

DT-Golog has been found to perform reasonably well compared to plain MDP solving. But how does it perform with our goal models? To explore this we tried it with different sizes of goal models, which we constructed by randomly combining smaller models we have developed for real domains (meeting scheduler, automatic teller machine, on-line bookstore and nursing). This way, the resulting artificial models preserved some degree of structural naturalness. Random numbers were entered for the probability values.

We had DT-Golog compute optimal policies for each root goal. The search horizon was set to the maximum plan length the goal model can yield. We used an Intel(R) Core(TM)2 CPU T5500 1.67 GHz with 4.00 GB RAM under Windows 7 to

Table 2. Time (in sec) to find optimal solution

Nodes	Bound	Time									
10	4	0.0	30	8	0.07	45	19	95.3	60	16	4395
20	8	0.08	40	12	3.7	50	14	75.6	65	21	(*)

perform the experiments. In Table 2, the time to get the result is given in seconds with respect to the size of the goal model, (*) signifying non termination within an hour – the bound also indicates the maximum plan length the model can yield. For design time analysis, the tool seems to perform adequately well for sizes up to about fifty nodes. Note also the dependency of the performance on the maximum plan length. We are optimistic that these times will improve as more research is taking place on the matter of reasoning performance (e.g. [15]). It is important to point that the presence of a DT-Golog program restricts the state space to a subset that is meaningful for the domain at hand. This allows DT-Golog to reason much more efficiently than e.g. a plain MDP-based approach would. In the LAS case we described, for instance, the overwhelming space of $30^{300} \cdot 2^{300}$ possible states did not prevent DT-Golog from doing useful analysis.

6 Related Work

Probabilistic analysis of requirements has been a subject for some investigation the past few years. Notable is the work by Letier and van Lamsweerde [2], in which goal structures offer the basis for structuring probability density functions that constitute a measure of achievement of non-functional objectives. Genetic-algorithm based reasoning was further proposed to allow for selecting static solutions that optimize such measures [16]. Recently, these ideas were applied for supporting obstacle analysis [7]. Our framework is different in a number of ways including that it focuses on agent action and dynamic aspects of the solutions (policies/plans) in addition to choices in the goal hierarchy and that it systematically integrates separate measures of priority, utility and probability in a semi-formal manner.

Probabilistic model checking with MDPs has been proposed in PRISM [17] and successfully used in a variety of applications – albeit not yet in the context of goal modeling. One fundamental difference between the model checker and DT-Golog that makes the later more suitable for our particular purpose is the fact that DT-Golog readily allows us to specify complex actions as *programs* and evaluate alternative designs, which is crucial for requirements analysis. Thus, DT-Golog goes beyond the classic MDP approach, where only primitive stochastic actions are allowed and not programs composed from such actions. Other approaches for dealing with uncertainty in requirements engineering have focused on self-adaptive systems and follow a fuzzy logic based approach [18,19]. In comparison, we model probability and utility as separate measures, and focus on automated reasoning about optimal behaviours, in terms of both those measures. In addition, a wealth of proposals exist for reasoning about goal models [20]. In that line of work, however, whenever dynamic aspects of the domain are considered, analysis is deterministic and does not take uncertainty of action into account.

7 Concluding Remarks

We presented a decision-theoretic framework for modeling and reasoning about stakeholder goals and priorities in the presence of uncertainty. The framework is based on the recognition that optimal solutions for fulfilling stakeholder goals will not necessarily be executed as planned, but may fail due to human or system error or other unknown factors. Therefore, to allow for pragmatic design-time analysis, we must take uncertainty into account. This calls for rethinking the semantics of standard goal models that is used for reasoning about alternatives. The main contributions of this paper towards those directions are an approach to probabilistically extend goal models to allow for modeling agent actions with uncertain effects together with stakeholder utilities and priorities, as well as a way to translate them into a formal specification language that allows for evaluating alternative designs based on utility optimization. We also show how detailed analysis can be performed using this toolset. Differences of our proposal from the work already done in the area include a strong focus on dynamic/behavioural aspects of solutions (i.e. sequences of tasks) and allowing exploration of the interplay between priority, utility and probability.

For the future, we wish to work on the core of the DT-Golog reasoner to also allow searching for local optima with respect to probability thresholds, effectively allowing trade-offs between probability and expected utility. Further, empirical assessment of the reliability and accuracy of precise DT-Golog analysis (and the effort investment it takes) is a priority. Scalability is an issue to be investigated in such a context. In terms of scalability of the modeling process, our current sense is that, due to the modularity of the probability and utility specification process (each task and soft-goal has its own table), larger goal models should easily accommodate definition of effects and utilities. In terms of scalability of the automated reasoning, our early results are encouraging for small-to-medium practical models. Nevertheless, we still need to explore solutions with larger models, such as breaking the problem into sub-problems [21].

Acknowledgements. This work has been partially supported by the ERC advanced grant 267856 for a project titled “Lucretius: Foundations for Software Evolution”, April 2011–March 2016 (<http://www.lucretius.eu.>)

References

1. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. *Science of Computer Programming* 20(1-2), 3–50 (1993)
2. Letier, E., van Lamsweerde, A.: Reasoning about partial goal satisfaction for requirements and design engineering. In: Proceedings of the 12th International Symposium on the Foundation of Software Engineering (FSE 2004), Newport Beach, CA, pp. 53–62 (2004)
3. Liaskos, S., McIlraith, S., Sohrabi, S., Mylopoulos, J.: Representing and reasoning about preferences in requirements engineering. *Requirements Engineering Journal (REJ)* 16, 227–249 (2011)
4. Mikhail Soutchanski. High-Level Robot Programming in Dynamic and Incompletely Known Environments. PhD thesis, Department of Computer Science, University of Toronto (2003)

5. Boutilier, C., Reiter, R., Soutchanski, M., Thrun, S.: Decision-theoretic, high-level agent programming in the situation calculus. In: Proceedings of the 17th Conference on Artificial Intelligence (AAAI 2000), Austin, TX, July 30–August 3, pp. 355–362 (2000)
6. Yu, E.S.K., Mylopoulos, J.: Understanding “why” in software process modelling, analysis, and design. In: Proceedings of the 16th International Conference on Software Engineering (ICSE 1994), Sorrento, Italy, pp. 159–168 (1994)
7. Cailliau, A., van Lamsweerde, A.: A probabilistic framework for goal-oriented risk analysis. In: Proceedings of the 20th IEEE International Requirements Engineering Conference (RE 2012), Chicago, IL, pp. 201–210 (2012)
8. Liaskos, S., Jalman, R., Aranda, J.: On eliciting preference and contribution measures in goal models. In: Proceedings of the 20th International Requirements Engineering Conference (RE 2012), Chicago, IL, pp. 221–230 (2012)
9. Boland, P.J.: Statistical and Probabilistic Methods in Actuarial Science. Chapman & Hall – CRC Interdisciplinary Statistics (2007)
10. Karlsson, J., Ryan, K.: A cost-value approach for prioritizing requirements. IEEE Software 14(5), 67–74 (1997)
11. Regan, K., Boutilier, C.: Robust policy computation in reward-uncertain MDPs using non-dominated policies. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010), Atlanta, GA (2010)
12. Regan, K., Boutilier, C.: Regret-based reward elicitation for Markov Decision Processes. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009), Montreal, QC, Canada, pp. 444–451 (2009)
13. Finkelstein, A., Dowell, J.: A comedy of errors: the london ambulance service case study. In: Proceedings of the 8th International Workshop on Software Specification and Design (IWSSD 8), Schloss Velen, Germany, pp. 2–4 (September 1996)
14. Pham, H., Soutchanski, M., Mylopoulos, J.: A simulator and a Golog implementation of the London Ambulance Service (LAS) Computer-Aided Despatch (CAD) system. Technical report, Department of Computer Science, Ryerson University (2006), <http://www.cs.toronto.edu/~mes/papers/LAS/index.html>
15. Böhnstedt, L., Ferrein, A., Lakemeyer, G.: Options in Readylog reloaded – generating decision-theoretic plan libraries in Golog. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007. LNCS (LNAI), vol. 4667, pp. 352–366. Springer, Heidelberg (2007)
16. Heaven, W., Letier, E.: Simulating and optimising design decisions in quantitative goal models. In: Proceedings of the 19th IEEE International Requirements Engineering Conference (RE 2011), Trento, Italy, pp. 79–88 (2011)
17. Hinton, A., Kwiatkowska, M., Norman, G., Parker, D.: Prism: A tool for automatic verification of probabilistic systems. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006. LNCS, vol. 3920, pp. 441–444. Springer, Heidelberg (2006)
18. Whittle, J., Sawyer, P., Bencomo, N., Cheng, B.H.C., Bruel, J.-M.: Relax: a language to address uncertainty in self-adaptive systems requirement. Requirements Engineering 15(2), 177–196 (2010)
19. Baresi, L., Pasquale, L., Spoletini, P.: Fuzzy goals for requirements-driven adaptation. In: Proceedings of the 18th IEEE International Requirements Engineering (RE 2010), Sydney, Australia, pp. 125–134 (2010)
20. Horkoff, J., Yu, E.: Analyzing goal models: different approaches and how to choose among them. In: Proceedings of the 2011 ACM Symposium on Applied Computing (SAC 2011), pp. 675–682. ACM, Taichung (2011)
21. Liaskos, S., Khan, S.M., Litoiu, M., Jungblut, M.D., Rogozhkin, V., Mylopoulos, J.: Behavioral adaptation of information systems through goal models. Information Systems (IS) 37(8), 767–783 (2012)

TBIM: A Language for Modeling and Reasoning about Business Plans

Fabiano Francesconi¹, Fabiano Dalpiaz², and John Mylopoulos¹

¹ University of Trento, Italy

{francesconi,jm}@disi.unitn.it

² University of Toronto, Canada

dalpiaz@cs.toronto.edu

Abstract. Conceptual models of different aspects of an organization—business objectives, processes, rules, policies and objects—have been used for organizational design, analysis, planning, and knowledge management. Such models have also served as starting points for designing information systems and conducting business intelligence activities. This paper proposes the Tactical Business Intelligence Model (TBIM), a language for modeling strategic business plans. TBIM lies in between the strategic and tactical level, for strategic plans are abstract business tactics. TBIM extends the BIM strategic modeling language with primitives for business model design. In addition to presenting the syntax of TBIM, we illustrate its usage through a medium-sized case study. We also propose a method for evaluating alternative plans through a mapping to business process models and the usage of simulation techniques.

Keywords: strategic planning, organizational models, business models.

1 Introduction

Organizations rely on a hierarchy of management layers, each focusing on different aspects of the organization. Conceiving an organization in terms of layers eases decision-making and management activities, for it refines the task at hand into smaller tasks at different levels of abstraction.

The topmost management level, called *strategic*, defines the direction of the organization. Notions such as vision, mission, and goal are essential components of a strategy. Once the strategy is set, a crucial decision is to be taken: *how* does the organization realize it? This question is answered by conducting *strategic planning* [1,11] activities, which lead to the definition of a high-level business *tactic* that, if implemented correctly, is expected to realize the strategy.

Strategic planning success depends on many factors, including the expertise of the management, the adoption of best practices, and the analysis of the key aspects of a business tactic (choosing the right ontology). While the topic has been widely explored in management science, there has been little work grounded on the usage of conceptual models to represent and analyze strategic plans.

In this paper, we propose the Tactical Business Intelligence Model (TBIM), a language for modeling strategic plans. The language is a link between the

strategic level and the tactical level in the sense that a business plan comprises a set of business goals, as well as the tactical plans for reaching those goals, defined in terms of value propositions, market segments, distribution channels, production and delivery activities, as well as partnerships. TBIM links these two levels by extending two state-of-the-art modeling techniques: (i) the Business Intelligence Model [10], a strategic modeling language based on primitives such as goal, situation, indicator, and (ii) the Business Model Ontology [18], which offers a core set of concepts to conceive business models at a tactical level.

Specifically, the contributions of the paper are as follows:

- We introduce the primitives of the TBIM modeling language. TBIM acts as a bridge between strategic and tactical models.
- We define a graphical notation for TBIM. The notation consists of two complementary views: the *tactical view* focuses on the internal aspects of a tactic, while the *partnership view* models the partnerships among enterprises.
- We provide a method for *reasoning* about alternative TBIM tactics through business process simulation techniques. Our method helps refining abstract TBIM tactics into more detailed tactics expressed as BPMN models.
- We illustrate our approach through snippets from a medium-sized case study concerning the organization of an international jazz festival [18].

Organization. Section 2 reviews our baseline. Section 3 introduces the TBIM language. Section 4 explains how TBIM models can be mapped to BPMN models. Section 5 shows the use of BPMN analysis techniques to evaluate alternative TBIM tactics. Section 6 discusses related work, while Section 7 concludes.

2 Baseline

Our baseline consists of BIM, a modeling language for strategic business modeling (Section 2.1), and a business ontology that defines the key factors to model a business tactic (Section 2.2). Our aim is to combine the set of modeling primitives provided by these approaches into a modeling language for strategic planning.

2.1 Business Intelligence Model (BIM)

The *Business Intelligence Model* (BIM) [10] is a modeling language for representing business strategies. BIM relies on primitives that decision makers are familiar with, such as goal, task/process, indicator, situation, and influence relations. BIM supports the notions from SWOT (*Strengths, Weaknesses, Opportunities, Threats*) analysis [4] by modeling internal and externals factors (situations) that are (un)favorable for fulfilling certain goals. BIM comes with automated reasoning techniques, including “what if?” and “is it possible?” analyses [10].

Figure 1 briefly illustrates the syntax of BIM by modeling part of the Montréal Jazz Festival (MJF) organization case study [18].

The top-level strategic goal is to Organize MJF Festival. To achieve this goal, five subgoals are to be pursued and fulfilled, including Provide attractive venue,

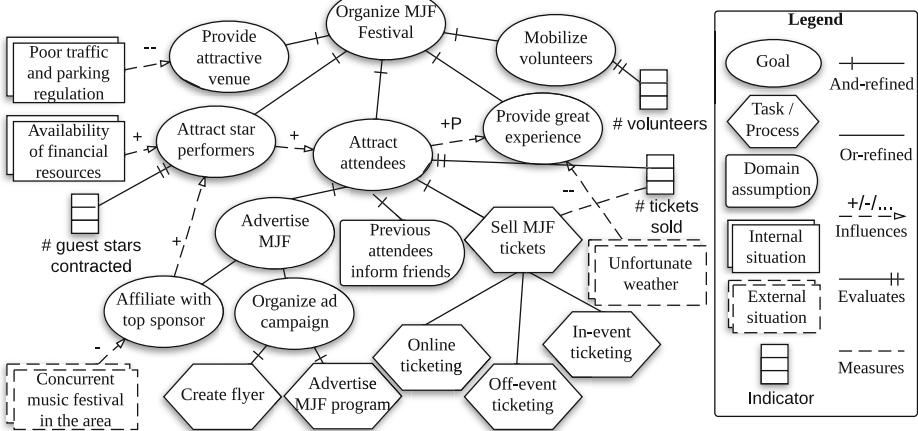


Fig. 1. Partial BIM model for the Montreaux Jazz Festival case study

Attract star performers, and Attract attendance. The latter goal requires to achieve goal Advertise MJF and to execute task Sell MJF tickets, assumed that Previous attendees inform their friends. The festival can be advertised by either affiliating with a top sponsor who also takes care of advertising, or internally organizing an ad campaign (which requires processes Create flyer and Advertise MJF Program). Goals and situations influence one another. The external situation Other music festival in the area influences negatively goal Affiliate with top sponsor, while the internal situation Availability of financial resources positively influences goal Attract star performers. Indicators are associated with goals so to evaluate to what extent the goal is fulfilled. For example, # volunteers evaluates goal Mobilize volunteers.

2.2 Business Model Ontology (BMO)

The Business Model Ontology (BMO) [18] argues for a set of success factors for e-business organizations. The proposed language proved to be very effective among practitioners, and has led to the creation of the renown *business model canvas* [19]. BMO is centered on four pillars:

- **Product innovation** is achieved when the company defines a *value proposition* that effectively reaches one or more *customer segments* by offering novelty, lower prices, or customer relationship excellence.
- **Infrastructure management** describes the value system configuration to deliver the value proposition, which includes defining *partnerships* and carrying out activities that use, consume, and produce *resources*.
- **Customer relationship** needs establishing high-quality client relationships, and reaching different client segments via adequate *distribution channels*.
- **Financial aspect** is a crosscutting concern in every organization. Defining a right balance between the *revenue* model and the *cost* structure is essential for the survival of the organization in the market.

3 Tactical Business Intelligence Model (TBIM)

We present the metamodel and the graphical syntax of TBIM. TBIM combines the strategic modeling framework provided by BIM with key elements from BMO that support the following set of requirements for strategic planning:

- R1. *Market segments.* Products and services are typically made available to specific customer segments. The language should be able to define *what* products and services an organization offers and to *whom*.
- R2. *Cross-organizational relationships.* The success of a strategic plan heavily depends on the establishment and maintenance of a network of partnerships with other organizations.
- R3. *Distribution channels.* Products and services are distributed through different channels. The choice of a specific channel depends on the customer segment that is approached by the provider.
- R4. *Resources and value propositions.* In order to create value, organizations use, create, consume, and transform resources [25]. Value propositions are resources that are a source of revenue for an organization [18].

TBIM consists of two complementary modeling views. The *tactical view* (Section 3.1) uses an extended version of BIM to describe the strategy of the modeled organizations as well as the high-level tactic to fulfill their goals. The *partnership view* (Section 3.2) represents a network of contractually-related organizations. Together, these two views do model alternative business plans (Section 3.3).

3.1 Tactical View

The UML class diagram in Figure 2 presents the metamodel of the tactical view. The gray-colored classes are adopted from BIM. We illustrate the graphical notation through the TBIM tactical view diagram in Figure 3.

Agent and Role. *Agents* represent a concrete organization or person. An agent is an active entity that carries out actions to achieve goals by exercising its knowhow [28]. Agents are *intentional*, for they carry out activities to achieve their goals. *Roles* are an abstract characterization of the behavior of a social agent within some specialized context or domain of endeavor. The term *Actor* refers generically to an agent or a role (is-a relationship in Figure 2). In Figure 3, for example, MJF is an agent that represents the festival organizers, while *Local customer*, *Loyal customer*, etc. are roles representing different types of customer.

Unlike BIM, TBIM models consist of multiple actors. Consequently, BIM entities such as goals and tasks fall within the scope of a specific actor.

Resource and Value Proposition. *Resources* are anything of value for the company being modeled. A *resource* can be animate (e.g., human, animal, etc.) or inanimate (e.g., wood, chair, money, etc.). In Figure 3, *Blank papers* are resources for the agent *MJF*. TBIM also includes value propositions as a specialization of resources. A *value proposition* is the statement of benefits that are delivered by the firm to its external constituencies [3]. They differ from plain resources as they carry an intrinsic value for the company, and they form its primary source

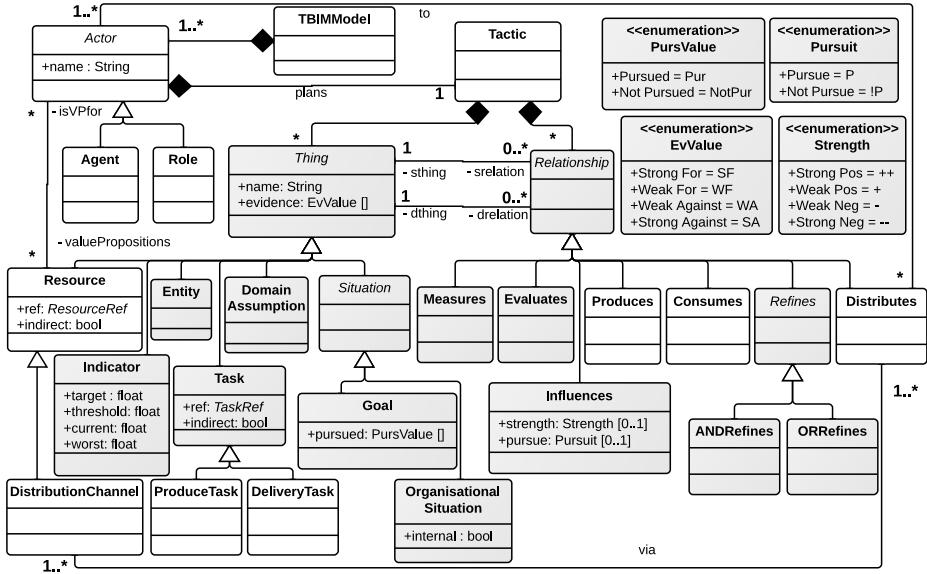


Fig. 2. Metamodel of the tactical view. Classes in gray are adopted from BIM.

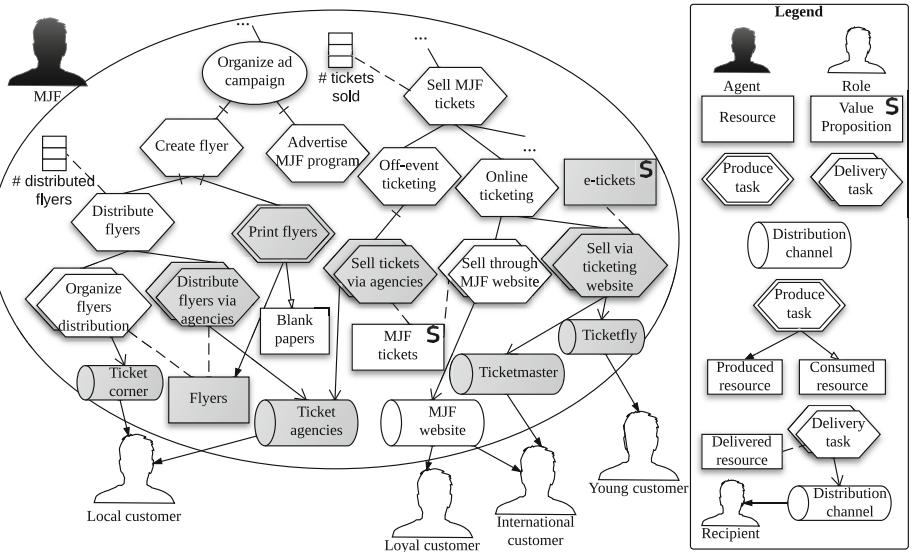


Fig. 3. Partial TBIM tactical view for the MJF case study. Gray-colored elements are indirect (obtained from other actors)

of revenue. For instance, MJF tickets are a value proposition for MJF, because their sales produce revenue for the festival. On the contrary, Flyers are a resource that does not carry revenue directly. TBIM distinguishes between *direct* entities (resources, tasks, distribution channels) that are internal assets of an actor, and *indirect* entities that are acquired from others via partnerships (see Section 3.2).

For example, **Flyers** are an indirect resource, meaning that MJF obtains them from another actor. In our graphical notation, indirect entities are gray-colored.

Produce Task, Produces, and Consumes. A *Produce Task* is a set of activities that results in the production of *resources*. A produce task is linked to produced resources via *Produces* relationships, and it can be connected to some resources via *Consumes* relationships, to indicate that the production process consumes those resources. Produce tasks specialize BIM tasks. While BIM tasks can decomposed to produce tasks, the latter type of tasks can not be decomposed, for their semantics is already very specific. In order to express that multiple produce tasks are needed, one can decompose a generic task into multiple produce tasks, each connected to an individual resource. In Figure 3, **Print flyers** is a produce task that consumes resource **Blank papers** and produces resource **Flyers**.

Distribution Channel and Delivery Task. *Distribution channels* are means through which customers are delivered resources. *Delivery tasks* are tasks indicating that resources are distributed to other actors. These tasks include the whole process of distributing a product, including packaging, shipment scheduling, and delivery. A delivery task is connected to at least one distribution channel. Multiple distribution channels can be associated with a delivery task to reach different market segments. Delivery tasks can not be further refined, but they can be refinements of a generic task. In Figure 3, **Sell through MJF website** is a delivery task, which encompasses the delivery of the value proposition **MJF tickets** through the channel **MJF website** to two types of customers: **Loyal** and **International**.

3.2 Partnership View

Maintaining an effective network of partnerships is key to the competitiveness of a company [9,18,24]. In TBIM, partnerships enable fulfilling strategic plans. Partnerships are stipulated through contractual agreements (*commitments*) that specify which products and services are made available, to whom, and in exchange for what. TBIM supports partnerships modeling through the partnership view. Its metamodel is shown in Figure 4 and illustrated in Figure 5.

Commitments. They are the principal element of the partnership view, and represent contractual agreements among actors on the execution of tasks, exchange of resources, and provision of distribution channels. A commitment [21] is a quaternary relation: a *debtor* actor commits to a *creditor* actor that a *consignment* will be delivered, if (optionally) a *reward* is provided by the creditor.

Commitments relate elements that appear in the tactical view: debtor and creditor are chosen among agents and roles, while resources, tasks (of all types), and distribution channels constitute the consignment and the reward. Since resources, tasks, and distribution channels appear (are contained in the tactical view within the scope of an actor, the metamodel of the partnership view includes references to those objects: *ResourceRef*, *TaskRef*, and *DistributionChannelRef*. The consignment and reward indicate the commitment of the involved actors to:

- *Resource provision*: a resource shall be transferred.
- *Task execution*: a generic process/task shall be carried out.

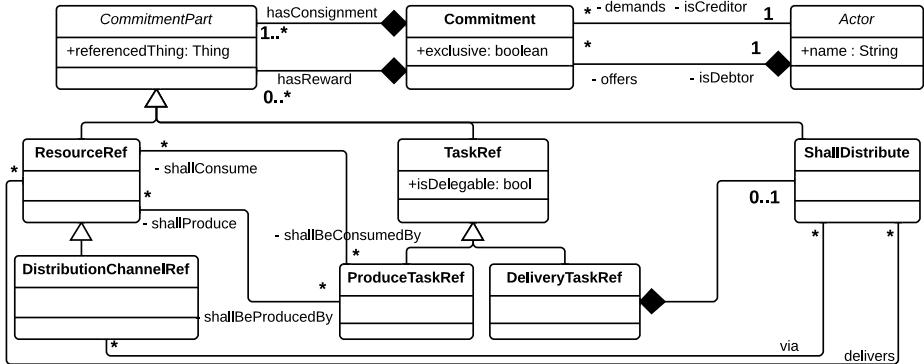


Fig. 4. Metamodel of the partnership view

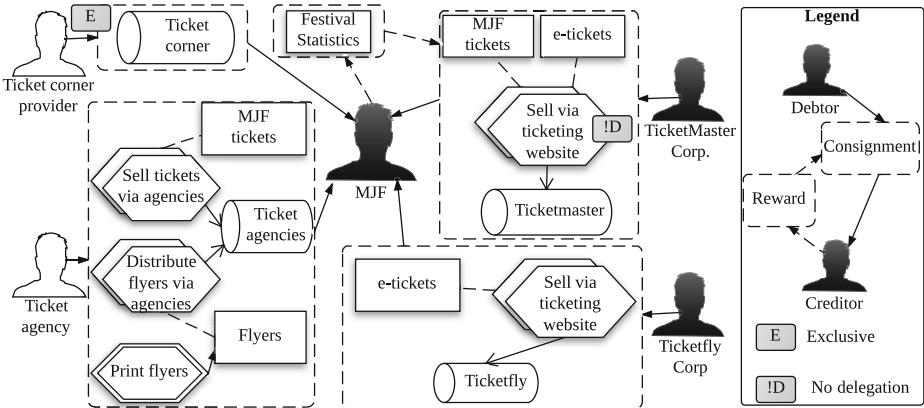


Fig. 5. Partial TBIM partnership view for the MJF case study

- *Produce task execution*: a production line is provided for producing resources.
- *Delivery task execution*: a delivery service for some items is made available.
- *Distribution channel provision*: a distribution channel is provided to enable a distribution process.

In Figure 5, role **Ticket agency** commits to agent **MJF** to execute produce task **Print flyers** (to produce **Flyers**), and delivery tasks **Distribute flyers via agencies** and **Sell tickets via agencies**. For both deliveries, the channel **Ticket agencies** will be used. The commitment from **TicketMaster Corp.** to **MJF** shows rewards: tickets are sold via the **Ticketmaster** channel only if **Festival Statistics** are provided from **MJF**.

A commitment defining a partnership can be further constrained:

- *ShallDistribute* (Figure 4) indicates that a delivery task shall deliver a specified set of resources via a specified set of distribution channels.
- A commitment may be *exclusive*, meaning that the consignment shall be provided to the creditor only. The commitment from **Ticket corner provider** is exclusive: the **Ticket corner** shall be used for selling **MJF** tickets only.

- A task reference in a commitment can be *delegable* (default) or not. If delegable, the debtor is authorized to delegate task execution to another actor. **TicketMaster Corp.** commits to not delegate task **Sell via ticketing website**.

3.3 Business Plans

TBIM allows representing alternative *business plans* to achieve the strategic business goals in the considered domain. A business plan is the process by which the entrepreneur, in exploiting an opportunity, creates a vision of the future and develops the necessary objectives, resources, and procedures (plans) to achieve that vision [20]. Business plans include value propositions, market segments, distribution channels, production and delivery activities, and partnerships [18].

Figures 3 and 5 show alternative business plans. Goal **Organize ad campaign** requires tasks **Create flyer** and **Advertise MJF program**. The former task requires the indirect production task **Print flyers**, which consumes **Blank papers** and produces **Flyers**, and is supported by the commitment from **Ticket agency** (Figure 5).

To distribute flyers, alternatives exist: either flyers distribution is organized internally, or the delivery task **Distribute flyers via agencies** is chosen. The latter alternative distributes **Flyers** to the market segment of **Local customers** through channel **Ticket agencies**. The commitment from **Ticket agency** supports this plan.

4 From Business Plans to Business Processes

The Business Process Modeling Notation (BPMN) [15] is the de-facto standard modeling language for business processes. BPMN models consist of activities (tasks and subprocesses) connected by a control flow. These models can be automatically analyzed to identify path execution times, bottlenecks, costs, etc.

BPMN modeling and analysis can be used to compare alternative TBIM tactics. To do so, we define a conceptual mapping between a TBIM model and a set of BPMN models. We assume that the indirect tasks and resources within the scope of an actor *A* appear in the consignment (reward) of at least one commitment made to (by) *A* by (to) actor *B*, and are also in the scope of *B*.

Actors. Every *agent* and *role* has to appear in at least one process as *pool*, as *lane* within a pool, or as *additional participant*. The same actor can be mapped to different elements and element types. For instance, actor **MJF** can be mapped to pool **MJF Administration** and additional participant **MJF Vice-President**.

Resources. Every direct *resource* that is produced or consumed by some task has to be mapped to at least one *data object*. In BPMN, a data object is information about what activities require to be performed and/or what they produce.

Tasks. *TBIM tasks*—of any type—describe conducted activities in an organization. Every direct task shall be mapped to at least one *BPMN task* in a BPMN process. These BPMN tasks shall appear in a pool or lane whose performer is the actor that owns the task in the TBIM model.

Resource Consumption and Production. Produce tasks do produce and consume resources. For direct produce tasks, we require that at least one corresponding

BPMN task has incoming and outgoing *data association* links to *data objects* that preserve the semantics of consumption and production, respectively.

Delivery. Delivery tasks denote routines for transferring a resource to another actor. For direct delivery tasks, we require the existence of a *message flow* from the pool/lane corresponding to the actor that owns the delivery task to the pool/lane corresponding to the recipient actor. *Distribution channels* are mapped indirectly (e.g., via tasks and/or messages), for BPMN has no primitive that carries the semantics of a distribution channel.

Commitments and Indirect Elements. The guidelines above take into account direct elements (tasks and resources). We examine now indirect elements (gray-colored, which appear in at least one commitment as consignment or reward). We show the case where the indirect element appears in the consignment. The mapping inverts debtor and creditor if the element is in the reward.

- *Resources* shall be modeled via BPMN *message flows* between pools or lanes. There should be at least one message from the debtor to the creditor where the *message* corresponds to the resource.
- *Produce tasks* shall be modeled as a two-way message flow: the creditor requests the production process, and the debtor provides the process outcome.
- *Distribution channels (without delivery task)* shall be modeled as a message flow from the debtor to the creditor, where the *message* is the provided channel (in TBIM, a distribution channel is a resource).
- *Delivery tasks* shall be modeled as a message from the creditor to the debtor, where the *message* requests the initiation of the distribution process.

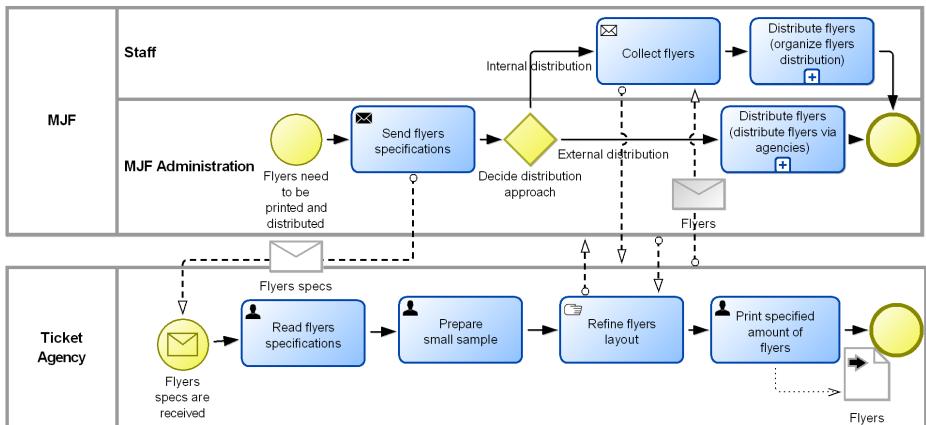


Fig. 6. Overall BPMN model for the distribution of MJF flyers

Figure 6 depicts a possible mapping of the *Distribute flyers* task refinement of Figure 3. In the topmost lane, MJF Administration sends flyers specifications to the Ticket Agency company, which provides flyers according to the partnership specifications (Figure 5). After some interactions, the process terminates with a

gateway to evaluate two different strategies (subprocesses): relying on an external distribution company, or handling distribution internally.

Our mapping defines compliance criteria between a set of processes and a TBIM model. The analyst can possibly derive process skeletons, but she would typically enrich them with fine-grained information, including additional BPMN tasks, different types of control flow, and structuring in subprocesses.

5 Evaluating Alternative TBIM Plans

Business process simulation enables evaluating processes and the alternatives therein in terms of execution time, usage of resources, and costs [23]. Simulation has been used for the analysis of organizations at design-time as well as in real-time environments as strategic and operational decision support tool [27].

We show how, given a set of processes for a TBIM model (Section 4), simulations can be run to compare alternative TBIM plans. TBIM models support alternative business tactics through OR-refinements and multiple partnerships for the same task or resource. To obtain such insights, we enrich the process models with information about cost, assigned resources, and execution times.

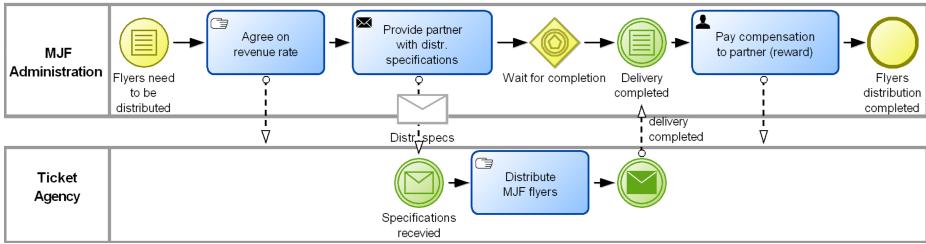


Fig. 7. BPMN model for the *external* organization of flyers distribution

Figure 7 shows the business process for distributing flyers through an external company (Ticket Agency). The process requires MJF Administration to agree on the revenue rate (a TBIM reward). Once agreed, the distribution is taken care of by the ticket agency, with no further involvement of the MJF administration.

Figure 8 shows a process for the internal organization of flyers distribution. MJF Administration hires people among the candidates provided by Temporary job agency. The candidates are interviewed and possibly hired. After a training period, the ticket corners are set up and provided with flyers, and the distribution starts. MJF administration copes with personnel sick leaves and resignations.

Enriching BPMN Models. Given a set of BPMN models created for a TBIM model using our guidelines, they need be enriched with information from the organizational context (e.g., BPMN can be assigned costs, execution times, and specific performers). While our mapping provides coarse-grained information about performers, the analyst may include more fine-grained details, specifying,

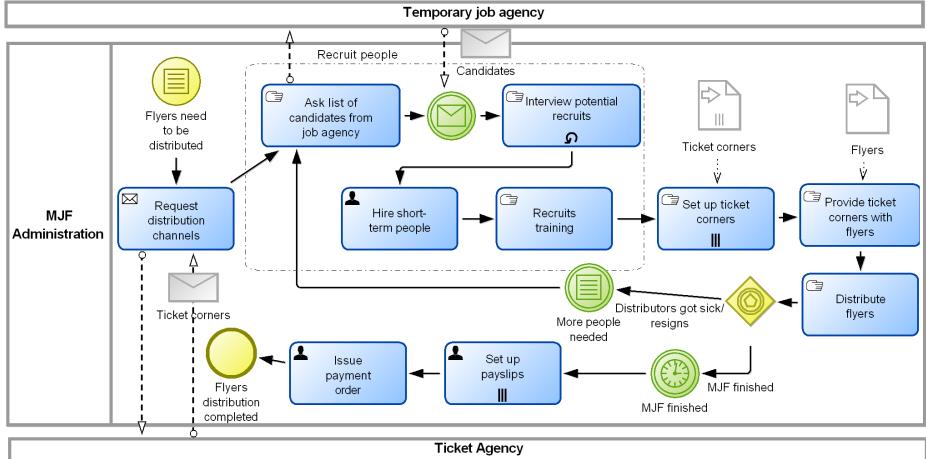


Fig. 8. BPMN model for the *internal* organization of flyers distribution

e.g., the specific person who is in charge for a task. This enables defining values for hourly wages and availability for each performer involved in a business process. We have enriched the BPMN models of Figures 6–8 with fictional values to illustrate our approach (details in our technical report [7]).

Table 1. Simulation results for the MJF flyers distribution process. Times are in business days, costs are in units.

Organizational Unit	Time		Pers. Costs		Other Costs	
	Ext.	Int.	Ext.	Int.	Ext.	Int.
Volunteer	0.0	1.71	0.0	0.0	0.0	0.0
Staff	0.37	3.55	30.0	145.71	0.0	100.0
MJF Administration	0.31	1.35	25.0	138.42	500.0	175.07
Recruits	0.0	17.25	0.0	1104.0	0.0	0.0
Ticket Agency	20.0	0.0				
<i>Total</i>	<i>20.68</i>	<i>23.86</i>	<i>55.0</i>	<i>1388.13</i>	<i>500.0</i>	<i>275.07</i>

Interpreting the Results. The results from the simulations include process costs, time spent in cycles, frequencies for each task, etc. We have applied the simulation component of the Adonis BPM toolkit [5] to our models. The results (Table 1) evidence that the internal approach is more time consuming. This is due to set-up activities for flyers distribution. The internal process is significantly more expensive in terms of personnel and other costs. Also, both staff and MJF administration would be relieved from some effort by relying on a partnership.

Improving TBIM Models. Simulation results can be used to ameliorate the TBIM tactic and/or to choose among alternative business plans (e.g., different tasks in an OR-refinement, or alternative partnerships). This activity relies on the expertise of the analysts. In our example, the results clearly suggest that

the partnership with a Ticket agency (Figure 5) is significantly better. The delivery task Distribute flyers via agencies is preferable over Organize flyers distribution (Figure 3), and channel Ticket corner is thus disposable/useless.

6 Related Work

We review related work about modeling different aspects of enterprises.

Business Ontologies. They define concepts to conceive enterprises. Two key approaches are Uschold’s enterprise ontology [24] and the Resource/Event/Agent generalized accounting model [14]. The Business Motivation Model [16] defines business plans by starting from the motivations of a company. These works provide sets of concepts (e.g., resources, duality, agents, strategy, activities, motivations) that are at the basis of several modeling languages, including TBIM.

Enterprise Architectures. They provide principles, methods, and models for the design and realization of an enterprise. TOGAF [17] promotes a requirements-centered design of the architecture, which begins with a vision of the architecture, and includes governance and change management. The Zachman framework [29] models enterprises by filling all the cells in a matrix where rows define the granularity level, and columns specify different aspects (why, when, what, how, where, who). These approaches do not offer a specific modeling language.

Business Modeling Languages. They represent different aspects of a business. The e³value [8] methodology models a network of enterprises creating, distributing, and consuming resources having an economic value. BMO and e³value share similar primitives [2]. Lo and Yu [13] suggest the usage of extended *i** [28] agent- and goal-oriented models to design collaborations—including resource exchange and task execution—among organizations. TBIM brings this notion further by suggesting different types of tasks (production, distribution), and uses commitments for relating business partners. *i** and e³value have been combined [9] to support e-service design. In their approach, the gaps between two models are filled in by the analyst. TBIM, instead, relies on a unified conceptual model.

Social Commitments. They are relationships that tie together autonomous participants through declarative contracts [21]. Telang et al. [22] rely on commitments to propose an agent-oriented approach for specifying and verifying cross-organization business models. TBIM relies on a more fine-grained ontology for both intentional elements and commitments.

Business Process Modeling (BPM). It is concerned with the creation of models of business processes. BPMN [26] is the de-facto standard notation for BPM, and relies on the notions of activity and control flow. BPMN 2.0 [15] introduces support for the collaboration between different organizations. We use business process models to analyze and evaluate alternative tactics. Our future work includes investigating the effectiveness of alternative BPM languages. An interesting candidate is the approach by Laurier *et al.* [12]. They simulate financial and operational performance through a mapping of concepts from the REA ontology to hierarchical, colored and timed Petri nets.

7 Discussion and Future Work

We have proposed TBIM, a conceptual modeling language for representing business plans. TBIM builds on the BIM language, and extends it with primitives from the BMO e-business ontology. We have also provided guidelines to map TBIM tactics to BPMN processes, and have shown how to use business process simulation techniques for evaluating alternative TBIM tactics.

A key feature of TBIM is that it decouples the internal tactics of an enterprise (*tactical view*) from the partnerships with other enterprises and customers (*partnership view*). This distinction enables determining if there exist unneeded partners, and if some tactical choice is not supported by any partnership.

Evaluation. We have illustrated TBIM and the usage of business process simulations with snippets from the MJF case study. Extensive models and results are available in our technical report [7].

Implementation. We have developed a proof-of-concept modeling tool to support the TBIM graphical notation. The tool is built using the meta-modeling development platform ADOxx [6]. This choice aims to facilitate integration with the Adonis BPM toolkit, which supports BPMN modeling and features sophisticated analysis and simulation algorithms.

Future work includes (i) improving the modeling tool to enable public use; (ii) developing features for automatically generating BPMN skeleton processes from TBIM models; (iii) evaluating TBIM on industrial case studies; and (iv) empirical evaluation of the language aimed at improving the modeling primitives.

Acknowledgments. This work has been partially supported by the ERC advanced grant 267856 for a project titled “Lucretius: Foundations for Software Evolution” (<http://www.lucretius.eu>), and by the Natural Sciences and Engineering Research Council (NSERC) of Canada through the Business Intelligence Network.

References

1. Abell, D.F.: Defining the business: the starting point of strategic planning. Prentice Hall (1980)
2. Andersson, B., et al.: Towards a reference ontology for business models. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 482–496. Springer, Heidelberg (2006)
3. Bagchi, S., Tulskie, B.: E-business models: integrating learning from strategy development experiences and empirical research. In: Proc. of the SMS Annual International Conference, pp. 15–18 (2000)
4. Dealtry, T.R.: Dynamic SWOT analysis: developer’s guide. Intellectual Partnerships (1992)
5. BOC Europe. Adonis BPM toolkit, <http://www.boc-eu.com/>
6. BOC Europe. ADOxx platform, <http://www.adoxx.org/>
7. Francesconi, F., Dalpiaz, F., Mylopoulos, J.: Tactical Business Intelligence Model (TBIM). Technical Report DISI-13-020, University of Trento (2013), <http://eprints.biblio.unitn.it/4148/4/techrep.pdf>

8. Gordijn, J., Akkermans, H., Van Vliet, J.: Designing and evaluating e-business models. *IEEE Intelligent Systems* 16(4), 11–17 (2001)
9. Gordijn, J., Yu, E., van der Raadt, B.: E-service design using i^* and e^3 value value modeling. *IEEE Software* 23(3), 26–33 (2006)
10. Horkoff, J., Borgida, A., Mylopoulos, J., Barone, D., Jiang, L., Yu, E., Amyot, D.: Making data meaningful: the business intelligence model and its formal semantics in description logics. In: Proc. of ODBASE, pp. 700–717 (2012)
11. Kaplan, R.S., Norton, D.P.: Having trouble with your strategy?: Then map it. Harvard Business School Publishing Corporation (2000)
12. Laurier, W., Poels, G.: Invariant conditions in value system simulation models. *Decision Support Systems* (in press, 2013)
13. Lo, A., Yu, E.: From business models to service-oriented design: a reference catalog approach. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 87–101. Springer, Heidelberg (2007)
14. McCarthy, W.E.: The REA accounting model: a generalized framework for accounting systems in a shared data environment. *Accounting Review* 57(3), 554 (1982)
15. OMG. Business Process Modeling Notation (BPMN) v2.0. Technical report (2006)
16. OMG. Business Motivation Model Specification v1.1. Technical report (2010)
17. Open Group. TOGAF Version 9. The Open Group Architecture Framework (2009)
18. Osterwalder, A.: The Business Model Ontology. PhD thesis, HEC Lausanne (2004)
19. Osterwalder, A., Pigneur, Y.: Business model generation—a handbook for visionaires, game changers, and challengers. Wiley (2010)
20. Sexton, D.L., Bowman-Upton, N.B.: Entrepreneurship: creativity and growth. Macmillan, New York (1991)
21. Singh, M.P.: An ontology for commitments in multiagent systems: toward a unification of normative concepts. *Artificial Intelligence and Law* 7, 97–113
22. Telang, P.R., Singh, M.P.: Specifying and verifying cross-organizational business models: an agent-oriented approach. *IEEE Transactions on Services Computing* 5(3), 305–318 (2012)
23. Tumay, K.: Business process simulation. In: Proc. of the Winter Simulation Conference, pp. 55–60. IEEE (1995)
24. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The enterprise ontology. *The Knowledge Engineering Review* 13(1), 31–89 (1998)
25. Wernerfelt, B.: A resource-based view of the firm. *Strategic Management Journal* 5(2), 171–180 (1984)
26. White, S.A., Miers, D.: BPMN modeling and reference guide. Future Strategies Inc. (2008)
27. Wynn, M.T., Dumas, M., Fidge, C.J., ter Hofstede, A.H.M., van der Aalst, W.M.P.: Business process simulation for operational decision support. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 66–77. Springer, Heidelberg (2008)
28. Yu, E.: Modelling strategic relationships for process reengineering. PhD thesis, University of Toronto (1996)
29. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal* 26(3), 276–292 (1987)

Automated Reasoning for Regulatory Compliance

Alberto Siena¹, Silvia Ingolfo¹, Anna Perini²,
Angelo Susi², and John Mylopoulos¹

¹ University of Trento, via Sommarive 14, Trento, Italy

{a.siena,silvia.ingolfo,jm}@unitn.it

² FBK-Irst, via Sommarive 18, Trento, Italy

{perini,susi}@fbk.eu

Abstract. Regulatory compliance is gaining attention from information systems engineers who must design systems that at the same time satisfy stakeholder requirements and comply with applicable laws. In our previous work, we have introduced a conceptual modelling language called Nòmos 2 that aids requirements engineers analyze law to identify alternative ways for compliance. This paper presents an implemented reasoning tool that supports analysis of law models. The technical contributions of the paper include the formalization of reasoning mechanisms, their implementation in the NRTool, as well as an elaborated evaluation framework intended to determine whether the tool is scalable with respect to problem size, complexity as well as search space. The results of our experiments with the tool suggest that this conceptual modelling approach scales to real life regulatory compliance problems.

Keywords: Conceptual Modeling, Automated Reasoning, Experimental Evaluation, Regulatory Compliance.

1 Introduction

The risk of information system non-compliance with relevant laws is gaining increasing attention from government and business alike, partly because of potentially staggering losses and partly because of growing public concern that, somehow, information systems need to be reined in. This trend has made regulatory compliance of software systems an important topic for Software and Information Systems Engineering: systems must comply with applicable laws (legal norms), in addition to fulfilling stakeholder requirements.

To deal with the problem of regulatory compliance, we need formal models of law that can be formally analyzed through various forms of reasoning to help requirements engineers find compliant solutions. Modeling approaches intended for law have been studied for decades in AI (more precisely, AI and Law), generally grounded on expressive, often modal, logics. Other approaches, grounded in Natural Language Processing and Information Retrieval, support different forms of analysis such as determining case similarity and relevance. We contend that

neither heavy-handed logical representations, nor natural language ones properly support the analysis requirements engineers need when they tackle the problem of regulatory compliance. Instead, we propose to use conceptual models of law that sit somewhere between logical and natural language models with respect to complexity, to help requirements engineers answer questions such as “In situation S, what are my alternative ways for complying with law fragment L?”, and if stakeholders have given preferences in addition to requirements, “What is a preferred compliance solution for law L, given situation S?”

The modeling framework for building conceptual models of law and capturing preferences (named Nòmos 2) has been presented in two companion papers [9,20]. This paper focuses on tool support for Nòmos 2. Given the size of law models, tool support is essential for any type of analysis. Accordingly, we have implemented such a tool that is founded on an inference engine (DLV [2,11]) to answer questions concerning compliance solutions in different situations, taking into account stakeholder preferences.

The specific questions addressed in this paper are: (1) What kind of automated reasoning is useful in tackling the compliance problem? (2) Can we have a reasoning tool that scales with the size and/or complexity of law models? In order to answer these questions, we first formalized reasoning mechanisms in terms of axioms, then conducted a series of experiments with the implemented tool. Artificial models of increasing sizes and with different properties were generated automatically and analyzed by the tool. The performance data from a series of runs were collected and analyzed. Our conclusions suggest that indeed the NRTool scales to problems of moderate law size.

The rest of the paper is organized as follows. Section 2 recalls the Nòmos 2 modeling language. Section 3 details the formal framework for reasoning on law models. Section 4 presents the NRTool that answers compliance queries through automated reasoning. Section 5 evaluates the efficiency of the tool through a scalability analysis involving a series of experiments. Section 6 surveys the state of the art and related work, while Section 7 concludes.

2 Baseline

Nòmos 2 [20] is a modeling framework that aims at capturing the variability of law. Indeed, legal texts often contain both a set of norms and elements such as conditions, exceptions and derogations, which make different norms hold under different conditions. These elements define a variability space, intended as alternative ways to comply with the set of norms within the legal text. This trait is captured in Nòmos 2 by differentiating *applicability* and *satisfiability* values for norms, and by defining *compliance* on the bases of the two.

Specifically, a norm is defined as a 5-tuple (*type*, *hol*, *ctrpart*, *ant*, *cons*): *type* is the type of the norm (e.g., duty or right); *hol* is the *holder* of the norm, the role having to satisfy the norm; *ctrpart* is the *counterpart*, the role whose interests are helped if the norm is satisfied; *ant* is the *antecedent*, the conditions to satisfy to make the norm applicable; *cons* is the *consequent*, the conditions to satisfy

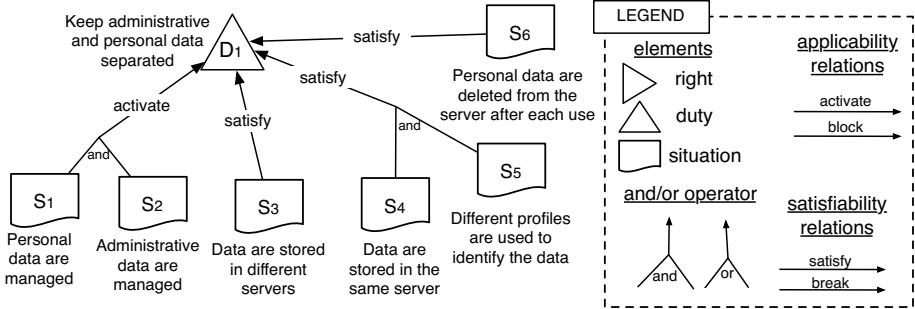


Fig. 1. An example of a Nòmos 2 model representing the duty for an administration office to keep the personal and administrative data of their clients/employee separated

in order to comply with the norm. Consequent and antecedents are modeled in terms of *situations*. A situation is a partial state of the world – or state-of-affairs – expressed through a proposition. For example, "Christmas season 2012" is a situation. A situation can be true, false, or have an unknown truth value. We use abbreviations ST, SF, SU to refer to truth values Satisfied True/False/Unknown, while AT, AF, AU refer to truth values Applicable True/False/Unknown. If the situations make the antecedent true, the norm applies; if the situations make the consequent true, the norm is satisfied. Situations are related to norms and to other situations by four basic relations. The *activate* relation, from a situation to a norm, means that if the situation is satisfied the norms is applicable; viceversa, the *block* relation makes the norm not applicable. The *satisfy* relation, from a situation to a norm or another situation, means that if the situation is satisfied the norm or the target situation is satisfied; viceversa, the *break* relation makes it not satisfied. Additionally, three shortcut relations have been defined between norms, in order to model the cases where one norm *derogates*, *endorses* or *implies* another one (see [20] for more details).

Depending on its applicability and satisfiability value, a norm may have value: *complied with*, *violated*, *tolerated* or *inconclusive*. For example, Figure 1 shows an example of a Nòmos 2 model describing the duties of an administration office that should keep personal data of their employees/clients separated from the administrative data used for running their business. When the two situations s_1 and s_2 hold (both have label ST), the conjunction of their labels is processed by the *activate* relation that propagates an applicability value to the norm (the label AT). Should s_3 hold, the duty will receive a label indicating that it is satisfied (ST). Since there is evidence that the duty is both applicable and satisfied, we say it is *complied*. Should there be no evidence of satisfiability for any of the situations that are linked with a satisfy relation (s_3-s_6 have label SU), the duty would be applicable and not satisfied — i.e., *violated*.

In a Nòmos 2 model when several relations target the same situation or norm (e.g., $s_3 \xrightarrow{\text{sat}} D_1$, $(s_4 \text{ and } s_5) \xrightarrow{\text{sat}} D_1$, $s_6 \xrightarrow{\text{sat}} D_1$), the values propagated to that target are treated as being in disjunction. So, as we can see in the example, different sets of situations can satisfy the duty D_1 and make it *complied with*.

Table 1. Axiom schema for invariants and propagation rules in norm models

Nr.	Description	Axiom	Definition
A1	prioritization for satisfiability	$ST(\phi) > SU(\phi) > SF(\phi)$	$ST(\phi) \wedge SU(\phi) \rightarrow ST(\phi),$ $ST(\phi) \wedge SF(\phi) \rightarrow ST(\phi),$ $SF(\phi) \wedge SU(\phi) \rightarrow SU(\phi),$ $ST(\phi) \wedge SF(\phi) \wedge SU(\phi) \rightarrow ST(\phi)$
A2	prioritization for applicability	$AT(\phi) > AU(\phi) > AF(\phi)$	$AT(\phi) \wedge AU(\phi) \rightarrow AT(\phi),$ $AT(\phi) \wedge AF(\phi) \rightarrow AT(\phi),$ $AF(\phi) \wedge AU(\phi) \rightarrow AU(\phi),$ $AT(\phi) \wedge AF(\phi) \wedge AU(\phi) \rightarrow AT(\phi)$
A3	default satisfiability value default applicability value		$\phi \neg ST(\phi) \wedge \neg SF(\phi) \rightarrow SU(\phi)$ $\phi \neg AT(\phi) \wedge \neg AF(\phi) \rightarrow AU(\phi)$
A4	compliance rule		$\phi AT(\phi) \wedge ST(\phi) \rightarrow Com(\phi)$
A5	not applicability rule		$\phi \neg AT(\phi) \rightarrow Tol(\phi)$
A6	compliance subsumption		$\phi Com(\phi) \rightarrow Tol(\phi)$
A7	duty violation		$\phi AT(\phi) \wedge \neg ST(\phi) \wedge Duty(\phi) \rightarrow Vio(\phi)$
A8	right non-exercitation		$\phi AT(\phi) \wedge \neg ST(\phi) \wedge Right(\phi) \rightarrow Tol(\phi)$
A9	inconclusiveness		$\phi \neg Tol(\phi) \wedge \neg Vio(\phi) \rightarrow Inc(\phi)$
	Description	Relation	Axiom
A10	satisfy	$\phi \xrightarrow{\text{satisfy}} \psi$	$ST(\phi) \rightarrow ST(\psi)$
A11	break	$\phi \xrightarrow{\text{break}} \psi$	$ST(\phi) \rightarrow SF(\psi)$
A12	activate	$\phi \xrightarrow{\text{activate}} \psi$	$ST(\phi) \rightarrow AT(\psi)$
A13	block	$\phi \xrightarrow{\text{block}} \psi$	$ST(\phi) \rightarrow AF(\psi)$
A14	and-satisfy	$(\phi_1 \wedge \phi_2) \xrightarrow{\text{satisfy}} \psi$	$ST(\phi_2) \wedge ST(\phi_1) \rightarrow ST(\psi)$
A15	and-break	$(\phi_1 \wedge \phi_2) \xrightarrow{\text{break}} \psi$	$ST(\phi_2) \wedge ST(\phi_1) \rightarrow SF(\psi)$
A16	and-activate	$(\phi_1 \wedge \phi_2) \xrightarrow{\text{activate}} \psi$	$ST(\phi_2) \wedge ST(\phi_1) \rightarrow AT(\psi)$
A17	and-block	$(\phi_1 \wedge \phi_2) \xrightarrow{\text{block}} \psi$	$ST(\phi_2) \wedge ST(\phi_1) \rightarrow AF(\psi)$

In order to select one (or a few) way of complying, out of many possible ones, we have extended our framework with a preference relation between situations [9]. The problem of compliance becomes then the *Preferred Compliance Problem* — i.e., the problem of finding a compliant solution to a norm model, given a set of applicable norms, such that the chosen solution best fits stakeholder preferences.

For example in Figure 1, deleting the data every time after each use can be considered a task more time consuming than using different server: we say that s_6 is less desirable than s_3 according to the time criterion ($s_6 <_{time} s_3$). The use of different server profiles in one server can be evaluated at least as desirable as using different servers: ($s_5 \leq_{time} s_3$). However, from an economical perspective, using two servers is more expensive than using one ($s_3 <_{cost} s_4$). Given all the preferences above, possible solutions to the norm model are evaluated and ranked.

3 Formal Analysis of Norm Models

In order to be analyzed, Nòmos 2 models need to be translated into sets of FOL formulas. Formally, a *norm model* is a pair $\{\mathcal{P}; \mathcal{R}\}$ where \mathcal{P} is a set of propositions and \mathcal{R} is a set of relations over \mathcal{P} . If $(\phi_1; \dots; \phi_n) \rightarrow \phi$ is a relation in \mathcal{R} , we call $\phi_1 \dots \phi_n$ source propositions and ϕ the target proposition of the relation.

Axioms. In Table 1 we introduce the axioms used to formalize the propagation of satisfiability and applicability values. We use six primitive predicates over propositions: $ST(\phi)$, $SF(\phi)$, $SU(\phi)$, $AT(\phi)$, $AF(\phi)$, $AU(\phi)$, meaning respectively that there is evidence that a given proposition ϕ is satisfied ($ST(\phi)$), not satisfied ($SF(\phi)$) or its satisfaction is undefined ($SU(\phi)$), that its applicability is true ($AT(\phi)$), false ($AF(\phi)$) or undefined ($AU(\phi)$). We establish a total order over satisfiability predicates $ST(\phi) > SU(\phi) > SF(\phi)$, meaning that $x > y \rightarrow \{(x \wedge y) \rightarrow x\}$; e.g., if there is conflicting evidence over ϕ , say $ST(\phi)$ and $SF(\phi)$ then $(ST(\phi) \wedge SF(\phi)) \rightarrow ST(\phi)$. Similarly, we have a total order over applicability predicates: $AT(\phi) > AU(\phi) > AF(\phi)$. Axioms A4–A9 state the four derived predicates for compliance. $Com(\phi)$ indicates that ϕ is *complied with*, being applicable ($AT(\phi)$) and satisfied ($ST(\phi)$). A tolerated norm ($Tol(\phi)$) is also complied with (Tol subsumes Com). A first case of tolerated norm is when the norm is not applicable (A5). Another tolerated case is when a right is applicable but not satisfied (A8) (e.g., a subject having a right but not exercising is a tolerated case). A violation ($Vio(\phi)$) is a case of an applicable duty that is not satisfied (A7). When none of the 3 cases above applies (compliance, tolerance or violation), we say that a norm is inconclusive (A9). Relation axioms define how the relations in a norm model propagate labels in order to deduce primitive and derived predicates. Satisfy/break relations define how positive/negative satisfiability values are propagated (A10, A11). Activate/block define how positive/negative applicability values are propagated (A12, A13). If neither a positive or negative value is propagated, an undefined value is propagated by default (A3). Axioms A14–A17 are used to characterize satisfiability/applicability values in case of a conjunction of values. The axioms for disjunction are defined similarly. For more details see [8].

Different propositions may be preferable over others. To capture this information, we add a set of binary reflexive, antisymmetric and transitive relations $\leq_C \in \mathcal{P} \times \mathcal{P}$, each \leq_C defining a partial order on propositions. Informally, we call these relations *preference relations*, and we read “ $\phi \leq_C \psi$ ” as “ ψ is at least as preferred as ϕ according to criterion C ”. We let “ $\phi =_C \psi$ ” abbreviates “ $\phi \leq_C \psi$ and $\psi \leq_C \phi$ ”, so that “ $\phi <_C \psi$ ” abbreviates “ $\phi \leq_C \psi$ and not $\phi =_C \psi$ ”. Informally reads “ ψ is strictly more desirable than ϕ according to criterion C ”. Each criterion C defines a partial order over propositions. Preference relations allow us to record relative preference of stakeholders between propositions, according to different criteria for comparison. Let \mathcal{C} denote the set of all criteria. We can further add relations between criteria, to help comparisons. We can define a hierarchy of domain-specific criteria for comparison, such as, for example: criterion *cost* is an aggregate of criteria *production cost*, *infrastructure cost*, *transportation cost*, etc. Such a structuring can help define aggregation functions and/or procedures to automatically rank alternative sets of propositions.

We do not discuss how preferences are negotiated between stakeholders, since different stakeholders can have opposing preferences over the same criteria. Both the definition of aggregation functions of preferences over criteria, and the negotiation of conflicting preferences are outside the scope of this paper.

Table 2. Propagation rules for satisfiability propagation

RULES FOR SATISFIABILITY PROPAGATION
R1 <code>sat_evidence(X1,true) ← sat_evidence(X2,true), situation(X2), satisfy(X2,X1).</code>
R2 <code>sat_evidence(X1,undefined) ← ¬ sat_evidence(X2,true), situation(X2), satisfy(X2,X1).</code>
R3 <code>sat_evidence(X1,false) ← sat_evidence(X2,true), situation(X2), break(X2,X1).</code>
R4 <code>sat_evidence(X1,undefined) ← ¬ sat_evidence(X2,true), situation(X2), break(X2,X1).</code>

Propagation Rules. The formal semantics defined in this section allows the support of formal analysis on norm models. To do this we represent a Nòmos 2 model as a database of facts, and using a declarative logic programming language we have define propagation rules that implement the Nòmos 2 axioms of previous section.

For example the propagation rules for the satisfiability of a proposition specify that, if there is a satisfy (or break) relation between two propositions and the source proposition is satisfied, then the target is also satisfied (or not satisfied, respectively). As we can see in table 2, the propagation rules associated with these axioms (A10, A11) ensure that the correct label is propagated depending on the evidence of satisfiability for the source proposition (Rule R1, R3). When there is not evidence of such satisfiability, indeed both relations propagate undefined evidence of satisfiability (Rule R2, R4).

These propagation rules have been defined for all our axioms in accordance with the rules in [20].¹ Once these rules are encoded, the user can therefore query this database of facts and infer the truth values of axioms. In the following section we describe the architecture of a tool (called NRTool), which implements the rules and exploits the DLV framework [11] to compute and verify the norm models.

4 Automated Reasoning with Norm Models

The specification and analysis of Nòmos 2 models — formalized in the previous section — is supported by a tool called Norm-Reasoning Tool, or NRTool. With this tool we can perform automated bottom-up and top-down analysis of a Nòmos 2 model, as described in [20], in order to support the analyst answer questions about these models; e.g., what are the applicable norms? Do we comply with a set of norms? What are the alternative ways to comply with a set of norms? etc.

A preliminary evaluation of this reasoning tool and its assessment on a small part of a case study is presented in [9].

Figure 2 describes the overall behaviour of the tool. The tool takes a structured representation of a norm model as input, and converts it into a Datalog logic program. Datalog [1] is a first-order logic program for querying deductive databases. A Datalog program is a set of rules of the form $r:- l_1 \wedge \dots \wedge l_n$, where

¹ See the technical report [8] for the full details of all propagation rules.

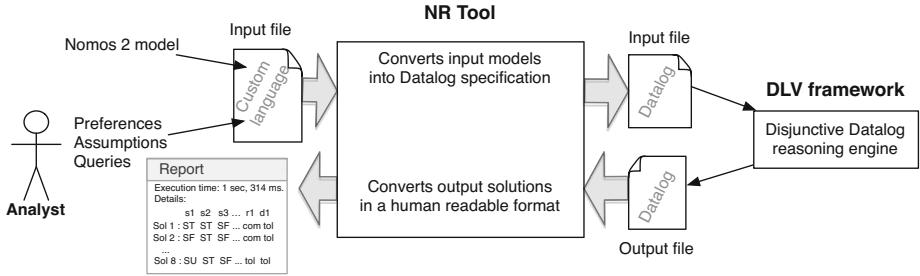


Fig. 2. Overall architecture of the NRTTool tool. It takes in input the Nòmos 2 model, the preferences, the assumptions and the queries from the analyst and transforms them into a disjunctive Datalog program as input for the DLV engine. The NRTTool reports the output of DLV to the analyst in a human readable format.

r (called the head of the rule) is a positive literal, and l_1, \dots, l_n are literals (called the body of the rule). Intuitively, the rule states that if l_1, \dots, l_n are true then r must be true.

The NRTTool relies on the DLV reasoning engine [2, 11] to execute the logic program and perform queries on the norm model. DLV is an Answer Set system that extends Datalog in different ways. It adds *disjunctions* in the rule heads, thus generating multiple alternatives; it adds support for true negations; it also supports weak constraints – i.e., constraints that can be violated at a cost, allowing solutions to be ranked according to the number of violations occurring. The search techniques and heuristics used by DLV are: backward search (similar to SAT algorithms), and advanced pruning operators, for model generation and innovative techniques for answer-set checking. DLV generates as output a complete set of models produced by the set of predicates and assignments to the variables or a pruned set of models that depends on input preferences.

Soft constraints allow us to identify the best compliance solution(s) in terms of their minimality. Since a large number of solutions could be returned – possibly too many – we are interested in having only the *best* solutions. To do this, we adopt the criterion of maximizing the number of not assigned values. The idea is that the fewer are the situations whose satisfiability is set to true or false in a compliance solution, the less analysts are constrained; viceversa, the more “undefined” situations we have, the more analysts are free to make their own decisions. By adding a soft constraint that situations’ satisfiability should be neither true nor false, we force the selection of the solutions with the highest possible number of “undefined” values.

The NRTTool maps situations and norms into Datalog facts, while relations are mapped into deduction rules. Moreover, the Nòmos 2 model is encoded via ground formulae (without variables and logical quantifiers). The output of DLV is parsed by NRTTool that presents it to the user in the form of a report specifying the truth value of the situations in the model and their respective compliance values.

Example: which set of norms is applicable? In this example we will see how the tool works in order to evaluate the applicable norms to a scenario. Answering to this question involves performing a forward reasoning analysis on the Nòmos 2 model. Given an initial values assignment for situations (expressed by the assumption), forward reasoning focuses on the propagation of these values to the norms accordingly to the propagation rules of Nòmos 2. The norms will receive applicability and satisfiability value depending on the relations in the law model. After translating this model into Datalog, the NRTool enables the reasoner to apply the propagation rule and calculate the applicability values. The NRTool then parses the output and returns to the analyst the set of norms of the model that are applicable.

5 Evaluation

Laws usually consist of tens or even hundreds of pages of natural language text, resulting in large models that may involve tens of thousands of concepts and links. In this section we investigate the scalability of our proposed reasoning tool with respect to the following three criteria (research questions):

- RQ1.** How does the tool scale with respect to the **size** of the problem, defined as the number of elements in the model?
- RQ2.** How does the tool scale with respect to the **complexity** of the problem, defined as the number of relationships constraining the different elements of the model? Also, how does the tool scale w.r.t. the number of solutions?
- RQ3.** How does the tool scale with respect to the **space of alternatives**, defined as the number of alternative refinements?

To answer these questions we have set up a testing framework, capable of producing artificial norm models with desired properties, run compliance analysis and record execution times. All experiments have been performed on an Intel i7 eight core 2.80 GHz computer equipped with 6 GB of memory running Linux version 2.6.18. The tool, the setting data, and the results generated by the experiments are available at <http://selab.fbk.eu/lawvariability/>.

5.1 Results

RQ1. To answer the first research question we have set up an experiment that tests the behaviour of the tool when the size of the norm model grows. A first model (the *input model*) was initially manually created. It consisted of 4 norms and 10 situations. Starting from this input model, 50 *test models* were then automatically generated. The generation algorithm consisted in: (i) creating a number (from 1 to 50) of replicas of the input model, resulting in models of size from 15 to 13875 nodes; (ii) creating a root norm that represents the full law; the root norm is and-refined into the root norm of each replica, through the *imply* relation; and (iii) adding a fixed number (10) of random relations from each replica to others. The number 10 was selected to ensure that our model has a

sufficient connectivity — indeed in our second experiment we study the impact of having different number of relations between replicas. The randomness of these relations was controlled by a parameter in our configuration called ‘seed’. It is worth noting that by changing this seed, the random relations also change, thus creating similar but not identical test models. The experiment was run 5 times with the same input model but different seeds.

The results of this experiment are reported in Figure 3. The figure reports on the x-axis the size of the model, expressed as number of nodes (including all types of Nòmos 2 concepts) of the test model. The y-axis reports the time taken at each execution to identify the set of solutions. There is a difference in one run, which results to be steeper, indicating a dependence of the slope from the seed; but the overall trend is quite linear in the considered problem size interval. In [20] we show an extract of a Nòmos 2 model for one column of HIPAA’s section 164.502, where we identified 15 situations. Given that the entire law consists of approximately 250 columns, an estimate of 4000 situations for the whole law is well within the boundaries of the models we tested.

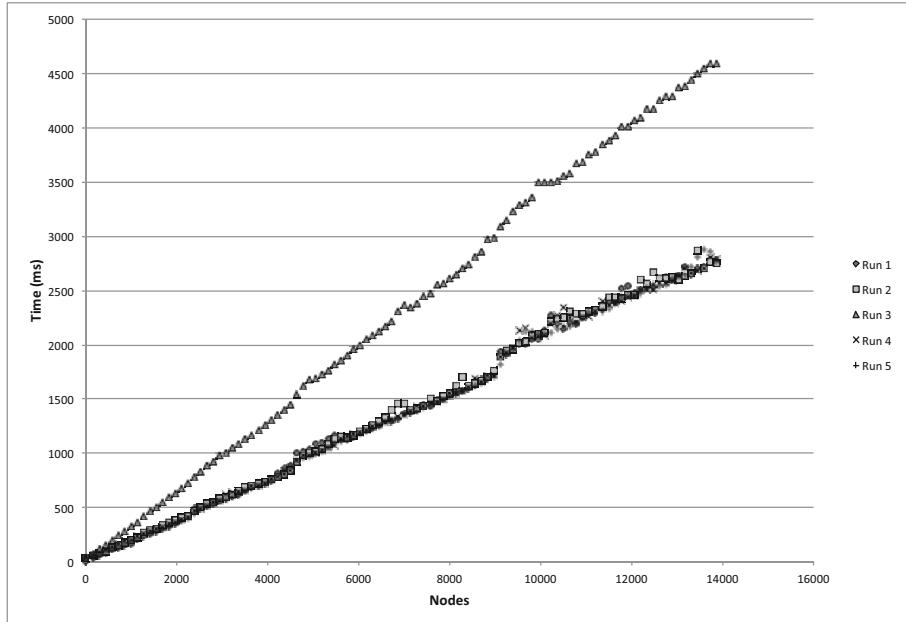


Fig. 3. Results from the experiments testing scalability wrt the size of the problem

RQ2. Answering the second research question requires understanding how the tool behaves when the connectivity of its input model changes. As in the previous case, we started from an input model and produced 700 test models. Differently from the previous case, now we kept fixed the size of the model, expressed as number of nodes, to a value of 225. Then, a random number of relations, varying from 0 to 750, were added to produce the test models.

The results of the experiments are reported in Figure 4. The x-axis reports the “connectivity” parameter — i.e., how many random relations have been added to the model. The y-axis reports the time taken at each execution to find the solutions. The left figure reports the execution time for all the connectivity values. A timeout of 60 seconds had been set, and in the first 7 runs the timeout was reached. For the remaining connectivity values execution time decreases significantly. The right figure magnifies the runs from a value of connectivity 8 to 500 and basically highlight the trend that is not possible to see in the left figure. In these cases the execution time decreases slightly and then increases again, smoothly. The reason of this behavior is that for unconstrained Nòmos 2 models (i.e., models with few relations among nodes) the number solutions depends exponentially on the number of nodes N (3^N , to be exact). As relations are added, the number of solution decreases, as does the time to find all of them. As more relations are added, the complexity of the problem to be solved — defined by the number of relations over a fixed graph — overtakes the cost of finding all solutions. This peculiarity results in the increasing trend shown on the graph of Figure 4. Besides this, we see a trend that decreases until a connectivity of 50, and then it increases smoothly.

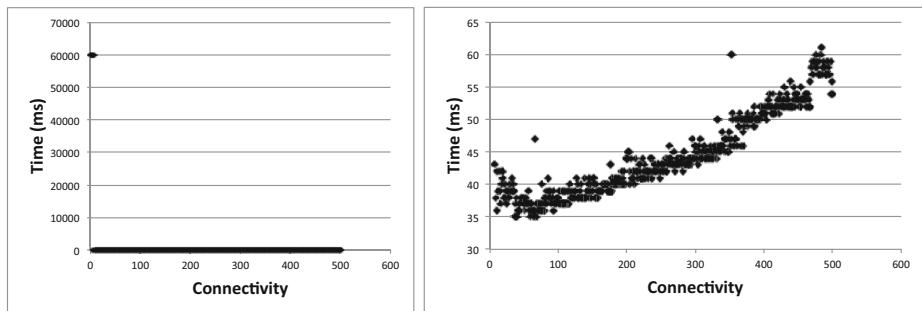


Fig. 4. Results from the experiments testing scalability wrt the complexity of the problem

RQ3. To answer the last research question we set up an experiment to analyze how the behaviour of the tool changes when specific constructs are introduced into the models. The constructs are those that theoretically change the number of available solutions, and so the space of solutions change without changing neither the size of the model nor its connectivity. The experiment was run with a model of 14000 nodes. 5 variations of the input model have been created. At each variation, the proportion between AND-relations and OR-relations has changed, from a value of OR of 0% (i.e., all the relations are in AND) to 100% (i.e. all the relations are in OR).

The results are shown in Figure 5. Here, the results are roughly the same with connectivity values of 0, 25% and 50%. With values of 75% and 100% (where the OR-decomposition becomes prevalent) times increase by approximately 20% passing from an average of 230 ms to 275 ms.

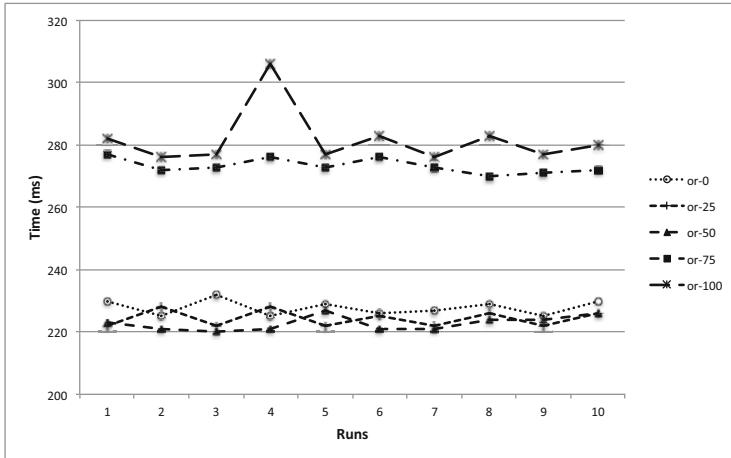


Fig. 5. Results from the experiments testing scalability wrt the space of alternatives

5.2 Discussion

The results of these experiments are twofold. On the one hand, we see a very encouraging linear trend for execution times, which generally corresponds to at most a few seconds. On the other hand, in the second experiment, we see in some cases times running out of bounds. This is due to the difference between *searching for a solution set* and *exploring the solution set*. The exploration time may overcome search time and diverge if the model is highly sparse and the space of alternatives is extremely large. Also with the third experiments we confirmed how the space of alternatives directly influences execution time. Moreover, as we can see from the second experiment, the initial constraints added to the model resulted first in a reduction of the time (as the number of solutions was decreasing) but then complexity kicks-in increasing overall execution times. The result of this experiment is comparable with similar investigations performed, e.g., in [17]. The lesson learnt from these experiments is that conceptual models can be a viable solution in analyzing laws for compliance, but only if the modelled laws are not too under-constrained. Given that laws are generally comprised of a high number of conditions, exceptions, derogations, cross-references and so on, we expect that real law models are not under-constrained.

6 Related Work

The main focus of this paper is the investigation of automated reasoning techniques to enable compliance analysis on real-size conceptual models of laws, including experimental evaluation of scalability. Similar approaches can be found in conceptual modelling for complex socio-technical systems [14], and for goal modelling in requirements engineering [7]. Relevant works on experimental evaluation of scalability of reasoning techniques and related tools, include the following.

In [11] the evaluation of DLV has been performed considering several problems, such as Traveling Salesperson or Quantified boolean formulas, having an increasing theoretical complexity (from NP to Σ_2^P). For each one of these problems, models of growing dimension have been considered; the performance of the solver has been measured in terms of the time necessary to solve the problems. In [17] is presented a method for randomly generating clausal formulae in modal logics. The paper describes several expected properties of good test sets, such as representativeness, reproducibility, parametrization, and presents the generating algorithms that produce 3CNF formula of growing complexity. Finally [5] characterizes hard SAT problems and identifies a “phase transition” in the problem attribute space.

In our work, we generate artificial norm models by cloning a manually defined input model. A similar procedure is used in [22], where a goal based framework for monitoring and diagnosing software requirements is presented. Thus, in our experimental evaluation we can generate models with increasing number of nodes (i.e. norms and situations), and increasing percentage of “AND/OR”, “activate/block”, and “satisfy/break” relationships, in a controlled way.

Worth pointing out that empirical evaluation in conceptual modelling has a wider scope with respect to what addressed in our paper, which, as reminded above, focuses on one specific but essential property to enable conceptual reasoning for regulatory compliance, namely scalability of automated reasoning. Addressing a different purpose with respect to ours in this paper, the empirical evaluation of conceptual models has been investigated from a domain understanding perspective. In this direction, [4] proposes a framework for the empirical evaluation of conceptual modeling grammars. [15] instead propose four criteria to evaluate conceptual modeling techniques. Differently from our work, these guidelines also focus on the effectiveness of the grammar modeled and the criteria to chose for its evaluation (independent variables, participants, ...). Recker [18] pursues a more philosophical-paradigmatic directions and discusses how existing evaluation methods can be assessed through the Bunge-Wand-Weber ontology. For a general overview on quality frameworks for conceptual modeling, Daniel L. Moody [12] presents a review of research in this field, identifies some theoretical and practical issues, and advocates the need of a common standard for the evaluation of quality of conceptual models.

Concerning the underlying approach in our work, namely law modeling for supporting compliance analysis, relevant related work are modeling approaches for RE and law, which extend existing RE modeling languages. For example, in [16] time line visualizations and decision trees are used to model legal terms or regulations in contracts. In [13] a Semantic Process Language (SPL) was created by combining Petri nets and a formal language, to describe legal regulations. In [21] business process models are checked for legal compliance through a modeling method called Event-driven Process Chain (EPC). However, all these approaches assess a different and specific aspect of legal compliance and appear therefore relatively isolated. [10] proposes a framework that supports analyzing the compliance of legacy Information Systems, which rests on the alignment of

a model of the transactions in the legacy system with an ontology of the laws that regulated the IS domain. This law ontology explicates the organizational roles, which correspond to the legal subjects of the laws governing the IS domain, with the domain artifacts and processes under their responsibility. Goal oriented techniques have also been used to represent legal prescriptions. For example, Darimont and Lemoine have used KAOS to represent objectives extracted from regulation texts [3]. Ghanavati et al. [6] use URN (User Requirements Notation) to model goals and actions prescribed by laws. Likewise, Rifaut and Dubois use i^* to produce a goal model of the Basel II regulation [19].

7 Conclusions

In this paper we have presented an implemented reasoning tool that supports situational analysis of law models. The technical contributions of this work include an axiomatic formalization of the reasoning mechanism realized by the tool, as well as its implementation based on an off-the-shelf inference engine (DLV). In addition, we report on a series of experiments that evaluated the tool for scalability with respect to problem size (the size of the model being analyzed), problem complexity (measured by the inter-connectivity of nodes in a model), and the space of alternatives (measured by the number of alternative refinements in a model). The results of these experiments suggest that the tool scales to real regulatory compliance problem involving a full law such as HIPAA.

The main limitation of our work is that our evaluations used artificial models. Accordingly, an important future research task will be the evaluation of the tool using Nòmos 2 models of real law. To this end, we need tools that support the generation of law models that are a good-enough approximation of a real law. Our future plans include exploring how to exploit existing tools for legal text analysis to support the extraction of Nòmos 2 models from text. Also, we plan to extend our language and reasoning tool to provide support for compliance analysis based on legal and social roles, delegations and related concepts.

Acknowledgments. This work has been supported by the ERC advanced grant 267856 “Lucretius: Foundations for Software Evolution” (April 2011 – March 2016) <http://www.lucretius.eu>.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Alviano, M., Faber, W., Leone, N., Perri, S., Pfeifer, G., Terracina, G.: The disjunctive datalog system DLV. In: de Moor, O., Gottlob, G., Furche, T., Sellers, A. (eds.) Datalog 2010. LNCS, vol. 6702, pp. 282–301. Springer, Heidelberg (2011)
3. Darimont, R., Lemoine, M.: Goal-oriented analysis of regulations. In: ReMo2V, held at CAiSE 2006 (2006)
4. Gemino, A., Wand, Y.: A framework for empirical evaluation of conceptual modeling techniques. Requirements Engineering 9, 248–260 (2004)

5. Gent, I.P., Walsh, T.: Beyond np: the qsat phase transition. In: Hendler, J., Subramanian, D. (eds.) AAAI/IAAI, pp. 648–653. AAAI Press / The MIT Press (1999)
6. Ghanavati, S., Amyot, D., Peyton, L.: Towards a framework for tracking legal compliance in healthcare. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 218–232. Springer, Heidelberg (2007)
7. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Goal-oriented requirements analysis and reasoning in the tropos methodology. *Eng. Appl. of AI* 18(2), 159–171 (2005)
8. Ingolfo, S., Siena, A., Jureta, I., Susi, A., Perini, A., Mylopoulos, J.: Reasoning with stakeholder preferences and law. research report. Technical report, University of Trento, Italy, TR-DISI-12-042 (2012), <http://selab.fbk.eu/lawvariability/>
9. Ingolfo, S., Siena, A., Jureta, I., Susi, A., Perini, A., Mylopoulos, J.: Choosing compliance solutions through stakeholder preferences. In: Doerr, J., Opdahl, A.L. (eds.) REFSQ 2013. LNCS, vol. 7830, pp. 206–220. Springer, Heidelberg (2013)
10. Khadraoui, A., Leonard, M., Thi, T.T.P., Helfert, M.: A Framework for Compliance of Legacy Information Systems with Legal Aspect. In: AIS Trans. Enterprise Sys. GIT Publishing GmbH (2009)
11. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.* 7(3), 499–562 (2006)
12. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering* 55(3), 243–276 (2005)
13. Olbrich, S., Simon, C.: Process modelling towards e-government - visualisation and semantic modelling of legal regulations as executable process sets. In: European Conference on E-Government (ECEG), pp. 405–414 (June 2007)
14. Paja, E., Dalpiaz, F., Poggianella, M., Roberti, P., Giorgini, P.: Sts-tool: Socio-technical security requirements through social commitments. In: Heimdahl, M.P.E., Sawyer, P. (eds.) RE, pp. 331–332. IEEE (2012)
15. Parsons, J., Cole, L.: What do the pictures mean? guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. *Data & Knowledge Engineering* 55(3), 327–342 (2005)
16. Passera, S., Haapio, H.: Facilitating collaboration through contract visualization and modularization. In: ECCE 2011, pp. 57–60. ACM (2011)
17. Patel-Schneider, P.F., Sebastiani, R.: A new general method to generate random modal formulae for testing decision procedures. *J. Artif. Intell. Res. (JAIR)* 18, 351–389 (2003)
18. Recker, J.C.: Conceptual model evaluation towards more paradigmatic rigor. In: CAiSE 2005 Workshops, pp. 569–580 (2005)
19. Rifaut, A., Dubois, E.: Using goal-oriented requirements engineering for improving the quality of iso/iec 15504 based compliance assessment frameworks. In: RE 2008, pp. 33–42 (2008)
20. Siena, A., Jureta, I., Ingolfo, S., Susi, A., Perini, A., Mylopoulos, J.: Capturing variability of law with Nōmos 2. In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012 Main Conference 2012. LNCS, vol. 7532, pp. 383–396. Springer, Heidelberg (2012)
21. Speck, A., Feja, S., Witt, S., Pulvermüller, E., Schulz, M.: Formalizing business process specifications. *Comput. Sci. Inf. Syst.* 8(2), 427–446 (2011)
22. Wang, Y., McIlraith, S.A., Yu, Y., Mylopoulos, J.: Monitoring and diagnosing software requirements. *Autom. Softw. Eng.* 16(1), 3–35 (2009)

Is Traditional Conceptual Modeling Becoming Obsolete?

Roman Lukyanenko and Jeffrey Parsons

Faculty of Business Administration, Memorial University of Newfoundland
{roman.lukyanenko, jeffreyp}@mun.ca

Abstract. Traditionally, the research and practice of conceptual modeling assumed relevant information about a domain is determined in advance to be used as input to design. The increasing ubiquity of open systems – characterized by heterogeneous and transient users, customizable features, and open or extensible data standards – challenges a number of long-held propositions about conceptual modeling. We raise the question whether conceptual modeling as commonly understood is an impediment to systems development and should be phased out for certain classes of information systems. We discuss the motivation for rethinking approaches to conceptual modeling, consider traditional approaches to conceptual modeling and provide empirical evidence of the limitations of traditional conceptual modeling. We then propose three directions for future conceptual modeling research.

Keywords: Conceptual Modeling, Information Systems Analysis and Design, Information Quality, Ontology, Cognition.

1 Introduction

Traditionally, information systems (IS) were developed and primarily used within organizational boundaries [1, 2]. IS development in this setting was *user- and consensus-driven*: users (or stakeholders) define system requirements and use and evaluate designed systems. Close proximity of analysts to users makes it possible for analysts to gather requirements, verify their fidelity, and resolve any conflicting perspectives before implementation. As users were mostly corporate employees or parties closely affiliated with the organization (e.g., suppliers), any individual or divergent views were generally subsumed by collective views.

The user/consensus-driven development perspective underlies prevailing approaches to *conceptual modeling* – a phase of IS development aimed at “formally describing some aspects of the physical and social world around us for the purposes of understanding and communication” [3]. Conceptual modeling conventionally results in specifications that capture relevant knowledge about the application domain. These specifications then guide development by supporting communication between developers and users, promoting domain understanding and guiding the design process [4].

The traditional modeling paradigm is increasingly challenged as more organizations draw on information outside organizational boundaries. In this environment,

it may no longer be feasible to reach all potential users and establish an agreed-upon specification of requirements for an IS. Prevailing approaches to conceptual modeling appear inadequate for modeling such heterogeneous distributed information. In this paper, we empirically examine if conceptual modeling as commonly understood is becoming an impediment for IS development in these settings.

2 Challenges to the Traditional Modeling Paradigm

Interest in distributed heterogeneous information is growing. Such information, for example, can better connect internal decision processes with information available to business partners, customers, and the general public [5-7]. Organizations are also looking to understand individual users (e.g., customers) to better cater to their unique needs. For example, personalizing product and service offerings increases sales [8].

Several recent technologies drive the need to effectively manage heterogeneous externally-generated information. The proliferation of mobile, miniaturized and ubiquitous computing exposes IS to diverse and unpredictable situations and demands systems be adaptive and flexible [9, 10]. The rise of semantic search engines (e.g., Facebook Graph Search), social networking and peer-to-peer computing (e.g., Facebook, Twitter, YouTube, Flickr) fuels demand for natural information exchange between people and machines. Of particular relevance is crowdsourcing, which engages users to work on sponsor-defined tasks [6]. Many crowdsourcing IS harness *unique* conceptualizations of diverse audiences. This is exemplified in a type of crowdsourcing known as citizen science that collects crowd data for scientific uses and broadly encourages creativity, and divergent thinking [5, 11].

Heterogeneous and distributed information poses a significant challenge to traditional conceptual modeling, which assumes that relevant aspects of reality are known or knowable in advance. The prevailing approaches to conceptual modeling, such as the Entity-Relationship (E-R) model and UML class diagrams, involve specification of conceptual entities (classes, entity types), attributes (or properties) and relationships between entities [12].¹ Developers use these models to produce IS design objects such as database schema, user interface, and code. IS use, including data creation, retrieval and manipulation, is then mediated by these objects.

As predominant conceptual structures are abstract in that they do not represent particular objects, but rather generalized or *stylized* [13] representations, the fundamental approach to conceptual modeling is *representation by abstraction* [14]. Abstraction-driven conceptual models deliberately ignore some aspects of reality, capturing only *relevant* information (where users or stakeholders indicate what is relevant). For example, a script made using the popular E-R grammar may depict entity types, attributes of entity types and relationship types with attributes. Entity types (e.g., *student, customer, equipment*) abstract from differences among instances (e.g., a *particular student*, or a *specific customer*), capturing their perceived equivalence.

¹ Constructs in other modeling approaches may include roles, actors, agents, goals, activities, processes, frames or patterns [14].

Representation by abstraction presupposes that consensus can be reached among users (stakeholders) on what is relevant. This assumption was considered unproblematic to the extent that development occurs in close contact with system users and other key stakeholders. Close contact with users provided an opportunity to resolve conflicts in individual views and generated an agreed-upon abstract conceptualization of a domain. This assumption is inherently limiting in heterogeneous environments. Aside from the difficulty of identifying and reaching all potential users in distributed settings, many emerging IS focus on capturing *unique* user views. In this context, imposing abstractions based on consultation with some users may marginalize, bias or exclude possibly valuable conceptualizations of other users [15, 16]. Recognizing shortcomings of traditional conceptual modeling, several alternative approaches to modeling dynamic, heterogeneous or distributed information have emerged.

One approach is to *reduce* the extent and depth of specifications. For example, models may employ only very basic concepts [17]. This concords with agile development which relies on lightweight (“barely good enough”) models that capture semantics minimally necessary for the next design iteration [18]. Here one challenge is to convey essential semantics while keeping models simple and lean [19].

Whereas lightweight modeling relies on a small number of “core” constructs, an alternative is to use grammars that capture *extended semantics*. Thus, extensions to popular conceptual modeling grammars have been motivated by the need to support dynamic information [20, 21]. For example, in dealing with unpredictability of heterogeneous information, such extensions may employ probabilistic classification models [22].

A growing interest is in *domain ontologies* that can “bridge” different systems and users [17]. These ontologies can be constructed by experts or be “outsourced” to the crowd thus purportedly generating more intuitive representations [23, 24]. Indeed, such approaches tend to encapsulate diverse user perspectives and are increasingly prolific. Yet even these models may neglect all valid views and inadvertently inhibit domain understanding [25]. Furthermore, domain ontologies generally require commitment of parties to a predefined (albeit often flexible) conceptual structure [17].

Another promising approach is putting the onus of modeling on users by allowing them to dynamically change models [10, 26]. This approach may be combined with lightweight modeling in which only a basic model is developed with the expectation that users update the model. This, however, invites unresolved issues of cooperative schema evolution and concurrent access and modification of schemas [26]. It is also unclear if this approach is reliable online, as some users may lack skills and motivation to create and alter models.

The approaches reviewed above presuppose some *a priori* structures and in this sense may have negative consequences similar to those in traditional modeling. Considering these and other concerns, there have been calls to develop novel modeling paradigms. Notably, Potts [27] examines *contextualism* in “which the particularities of the context of use of a system must be understood in detail before the requirements can be derived” (p. 102). Lukyanenko and Parsons [28] propose modeling principles for crowdsourcing that depict *concrete* (rather than abstract) instances and attributes.

In this paper we empirically examine whether it is advantageous to develop IS in heterogeneous environments without employing abstraction-driven grammars – we thus consider and evaluate the “*no modeling*” approach.² Under this approach, development proceeds by selecting a flexible data model and a flexible user interface. Users are then instructed to provide information according to their own conceptualization of reality without having to conform to a particular structure.³ Such information can be stored in a flexible data model such as instance-based [29], graph [30], or semi-structured [31] data model.

For example, using the instance-based data model, information can be collected without having to classify relevant instances; information about instances can be stored in terms of attributes [29]. Different users can supply different attributes for the same instance. Failure to agree on classes, relationship types or attributes is no longer problematic as both convergence and divergence of views is accommodated: any relevant attribute can be seamlessly captured. The attributes can be then queried to select instances stored based on classes of interest or other criteria. Thus, classes and other abstract constructs are not necessary before implementing such a system and conceptual modeling may not be needed for the design phase (at least not for the purposes of generating a database schema and other design elements).

Indeed, the instance-based or other flexible solutions appear to address the challenges of reaching consensus and accommodating individual and unanticipated uses. Critically these solutions make it possible to bypass a major part of IS development – the creation of a formal representation of knowledge in a domain. This significantly simplifies systems analysis and does so in an environment in which traditional conceptual modeling is problematic. Furthermore, the instance-based IS appears to improve data quality (e.g., accuracy per unit of data) and information yield (e.g., greater number of instances stored) compared to more traditional (i.e., class-based) systems [15, 32].

3 Experiment

To empirically evaluate a no modeling instance-based IS, we designed a laboratory experiment in the context of online citizen science.⁴

Many popular citizen science applications epitomize the modeling challenges discussed above. These systems are established primarily to serve the needs of scientists, but the actual users or contributors (i.e., citizen scientists) are ordinary people, often lacking subject matter expertise and possessing diverse domain views [32]. Imposing a particular view upon content creators may focus (or bias) contributors to one

² Here, we consider *no modeling* in a limited sense as denoting the absence of a traditional specification of the kinds of information that an IS is designed to manage. We recognize, however, that any development inherently involves some degree of modeling.

³ The authors are currently developing a real “no modeling” IS in the citizen science domain using the instance-based data model.

⁴ This experiment was also used to provide support for the impact of class-based models on data accuracy; this issue is beyond the scope of the current study and is reported elsewhere.

particular goal (e.g., species identification, classification of galaxies), but fail to capture additional information citizen scientists may wish to communicate.

Current approaches to citizen science follow traditional modeling principles. Popular citizen science projects (e.g., www.eBird.org, www.iSpot.org.uk) focus on positive identification of species or genera (e.g., American Robin). Species and genus are classification levels with widely accepted scientific *utility*. In contrast, the generally preferred classification level for non-experts is the *basic level* [33]. Unlike the species level, the basic level (e.g., bird, fish, tree) tends to be an intermediate taxonomic level (e.g., “bird” is a level higher than “American Robin”, and lower than “animal”).

Species/genus-level classes represent useful classes in a natural history application, while basic-level classes operationalize intuitive classes natural to non-expert users; therefore both are reasonable for constructing abstraction-driven conceptual models of the citizen science IS. To contrast traditional conceptual modeling with a “no modeling” alternative, we explore an instance-based solution to citizen science where sightings of organisms are reported in terms of attributes of instances [16]. Users are thus not required to comply with *a priori* created models of abstraction (e.g., classes).

Consistent with philosophy and cognition that postulate uniqueness of individual instances and mental models of instances [34, 35], we argue non-expert participants, if given the opportunity, will provide substantial numbers of unique attributes. Since abstractions such as classes are based on commonalities of instances, they will be unable to accommodate some of the attributes participants provide. Furthermore, as it may be difficult to *a priori* anticipate the attributes that are salient for all users, it is infeasible to choose classes that will account for all attributes. We thus hypothesize:

Hypothesis: *Non-experts will describe instances in terms of attributes that cannot be captured by definitions of classes (intuitive and useful) used to model instances.*

While we predict that many attributes provided by different users will be unique, it is also desirable to have some degree of attribute agreement. Indeed, complete disagreement (i.e., no overlap in attributes provided by different participants) would mean that using attributes to represent perceived reality is unreliable. To broadly ensure the value of collecting and storing attributes of instances, agreement on a core set of attributes should hold for both familiar (e.g., instance of American robin) and unfamiliar (e.g., instance of obscure mushroom) instances. Thus, we wish to investigate the degree to which non-experts converge on the attributes used to describe instances and seek to answer the following exploratory question:

Question: *Do non-experts demonstrate significant agreement on a core set of attributes of familiar and unfamiliar, complex and simple instances?*

We conducted a study among potential citizen scientists - 247 undergraduate business students (141 female, 106 male) at a Canadian university. The experiment was conducted in 8 sessions and the order of stimuli was randomized between sessions.

Business students were chosen to ensure low level of expertise in biology, reflecting the intended context where users are members of the general public. Low domain expertise was verified using self-reported expertise measures and more objective

measures: 83% of participants either strongly or somewhat disagreed (on a 5-point scale) with the statement that they are “experts” in local wildlife (mean=1.90; s.d.=0.886). More than two thirds of the participants (77%) had never taken any post-secondary courses in biology.

The stimuli were 24 full-color images of plants and animals (all different biological species) native to the geographic region in which the study was conducted. The stimuli were selected by an ecology professor expert in flora and fauna of the region. Species were chosen to include some organisms believed to be familiar and unfamiliar.

Participants were randomly assigned into one of two study conditions. Those in the “Categories and Attributes” condition (122 participants) were given a printed form with two columns - one asking participants to *name* the object on the image (using one or more words) and the second asking them to *list features that best describe* the object. In the “Attributes only” condition (125 participants), there was only one column asking participants to *list features that best describe* the object.

Once categories and attributes were entered, we coded categories as either “basic level,” “species-genus level,” or “other” and attributes as either “basic level,” “superordinate to basic,” “subordinate to basic,” and “other.” The species-genus level was determined based on biological convention, while the basic level was adopted from prior studies in cognitive psychology. A thorough survey of cognitive literature failed to reveal an agreed-upon basic-level for 6 out of the 24 species used as stimuli (lung lichen, Old Man’s beard, coyote, chipmunk, moose, and caribou). The final data set contained 25,315 records, with 6,397 categories and 18,918 attributes. The number of unique attributes and categories was 1,673, with 264 categories and 1,409 attributes.

We first provide evidence that non-expert participants generally do not prefer species/genus level to classify instances and these responses are generally not as accurate as more intuitive basic-level classes. To do this, we analyze categories in the “Categories and Attributes” condition. In this condition, 122 participants provided a total of 3,737 categories (an average of 1.28 per image per participant). As expected, participants prefer to classify using basic-level categories and these classifications tend to be more accurate than when attempting to classify at species/genus levels (see Table 1). The exceptions (i.e., American robin, Blue Jay, Killer Whale) appear to be common organisms that participants are frequently exposed to in nature or through media.

These results confirm the operationalization of basic-level as an intuitive class. This is critical in testing the extent to which participants employ basic-level attributes (e.g., *can fly, has feathers* for *bird*) versus lower-level attributes (e.g., *red breast*). The greater the number of sub-basic level attributes, the greater the extent to which a conceptual model built on the basic level omits all information non-experts are able to provide. To investigate these issues, the attributes (7,330) in the Attributes-only condition for the 18 plants and animals were classified into: sub-basic, basic (and superordinate), or other, resulting in 6,429 sub-basic, 824 basic, and 77 other attributes. As expected, in contrast with the prevalence of basic level categorization, there were significantly more sub-basic attributes (average *p*-value approaching zero, see Table 2). This suggests that using intuitive classes (which tend to be general for non-experts) in models prevents a considerable number of attributes from being captured.

Table 1. Number and accuracy of basic categories (BC) and species-genus categories (SG) (** -sig. at 0.01 level; ** -sig. at 0.05 level)

Common name	No of BC	No of SG	χ^2 No of BC vs. SG	Correct BC	Correct SG	Fisher's exact p-val. Accuracy of BC vs. SG
Blue W. Teal	144	5	129.67***	143	0	0.000
Mallard Duck	133	20	83.46***	133	15	0.000
Spt. Sandpiper	112	2	106.14***	112	0	0.000
Caspian Tern	111	2	105.14***	111	0	0.000
Red fox	110	14	74.32***	104	10	0.015
Labrador tea	108	4	96.57***	108	0	0.000
G. Yellowlegs	108	1	105.04***	107	0	0.018
Common Tern	107	3	98.33***	107	0	0.000
Red squirrel	105	18	61.54***	100	1	0.000
Sheep laurel	103	2	97.15***	103	0	0.000
Atl. Salmon	100	25	45.00***	100	0	0.000
Fireweed	94	26	38.53***	94	1	0.000
Calypso orchid	92	12	61.54***	91	0	0.000
Indian pipe	89	7	70.04***	88	0	0.000
Amer. Robin	86	78	0.39	86	74	0.049
Blue Jay	69	99	5.36**	69	98	1.000
Killer whale	54	88	8.14***	48	86	0.054
False morel	34	0	N/A	22	0	N/A

Table 2. Number of basic and subordinate attributes

Species	Sub-basic	Basic	Diff: χ^2 p-val	Other Attr.
American Robin	362	35	0.000	3
Atlantic salmon	273	45	0.000	19
Blue Jay	397	51	0.000	5
Blue Winged Teal	350	76	0.000	13
Bog Labrador tea	266	3	0.000	5
Calypso orchid	358	3	0.000	3
Caspian Tern	460	47	0.000	4
Common Tern	435	41	0.000	3
False morel	238	9	0.000	1
Fireweed	302	3	0.000	7
Greater Yellowlegs	486	39	0.000	9
Indian pipe	342	6	0.000	3
Killer whale	325	54	0.000	9
Mallard Duck	421	74	0.000	2
Red fox	340	46	0.000	90
Red squirrel	362	105	0.000	36
Sheep laurel	319	4	0.000	3
Spotted Sandpiper	393	44	0.000	1

We now evaluate the same hypothesis with respect to the species-level classes. Although we demonstrate low natural frequency of responses at that level, in principle it may be possible to design a user interface that guides users to species-level classes because they are valuable to project sponsors. We argue, however, even these specific classes would fail to account for all attributes. Thus, the greater the number of attributes not captured by species classes, the greater the degree to which a conceptual model built on species-level misses all information non-experts are able to provide.

We compare the attributes provided in the Attributes-only condition with attributes from the species identification guides considered standard for identifying at the species-level [36-39]. One of the authors matched each attribute provided by participants with attributes of the organism in the field guide. The comparison was based on approximate similarity (e.g., *gray underbelly* and *whitish underbelly* were considered equivalent), erring on the side of similarity (to increase conservativeness of the test).

As predicted, while many attributes provided can be inferred from classifying organisms at the species-level, participants provide significantly greater than zero number of attributes not accounted for by the applicable species class (see Table 3). Among those, some are *instance attributes* in that they describe a particular instance (e.g., *standing on rock*, *looking sick*, *dorsal fin is deformed*); some describe features considered not salient for identification at the species-level (e.g., *blue eyes* for American Robin, *black feet* for Blue Jay); some attributes are orthogonal to biological taxonomy (e.g., *weed-like*, *beautiful*, *scary*). Thus, modeling using specific species-level classes fails to account for a large proportion (49.0% of subordinate attributes) of attributes used by non-experts when describing common and uncommon instances.

Table 3. Number of subordinate, species-level and non-species-level attributes

Common Name	Sub-basic	Species	Non-species
American Robin	362	180	182
Atlantic Salmon	273	100	173
Blue jay	397	176	221
Blue W. Teal	350	156	194
Calypso Orchid	358	117	241
False morel	238	162	76
Fireweed	302	137	165
G. Yellowlegs	486	362	124
Indian Pipe	342	193	149
Mallard duck	421	238	183
Sheep Laurel	319	122	197
Sp. Sandpiper	393	221	172

Finally, we examine the question: “to what extent do non-experts agree on the attributes of familiar and unfamiliar phenomena?” Answering this question is important in determining whether data collection based on instances and attributes can generate consistent data. To address the issue, we assessed agreement on 9,556 attributes provided by 125 participants for all 24 organisms in the Attributes-only condition.

To evaluate agreement we employed the theoretically-driven approach of model testing and compared two hypothetical models. The null model represents the absence

of statistically significant agreement on a core set of attributes. Under the null model, some attributes may be used by more than one participant (e.g., due to limited domain vocabulary, rudimentary beliefs about a domain, or by simple chance); yet there is no “core” set of attributes that many participants agree on. The corresponding distribution of attribute frequencies is assumed to be uniform. The alternative model represents the hypothesized agreement among observers on core attributes for the observed instance. The alternative model should demonstrate, with statistical significance, a non-uniform distribution of attribute frequencies (e.g., Pareto distribution). Similar to the null, the alternative model may contain many idiosyncratic attributes with low frequencies, signifying individual perceptions of attributes of instances. Unlike the null model, however, it will also reflect a small number of highly frequent attributes reported by a large number of participants – demonstrating strong agreement on a small number of “key” attributes.

To test the two models, we computed maximum likelihood-ratio G-test. Here, the expected values are determined assuming the null model of uniform distribution and are obtained by taking the sum of all frequencies divided by the number of reported attributes. For example, participants provided 400 total and 85 unique attributes describing American robin (see Figure 1). The expected value for each attribute is 4.71 (which is less than 5, thereby justifying G-test technique). The resulting G-statistic was computed to be 772.11 ($p < 0.001$ with 84 d.f.). This procedure was repeated for the other 23 stimuli with similar results: all attribute frequencies were found to be non-uniformly distributed. The results were highly significantly with an average p -value approaching zero.

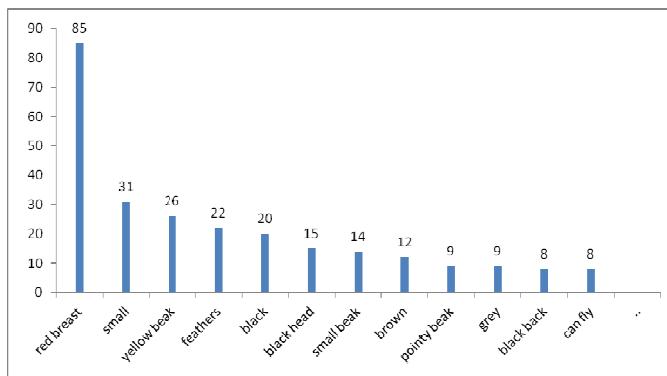


Fig. 1. Attributes for American robin in Attributes-only condition

These results indicate the attribute frequencies are not uniformly distributed, demonstrating statistical *agreement* among non-expert observers of familiar and unfamiliar, feature-rich and feature-poor (perceptively, based on the image) natural history instances in the study.

We proceeded with Kolmogorov-Smirnov and Anderson-Darling goodness-of-fit statistics to fit data to common distributions. While different distributions exhibited better fit for different species, the general families of better fitting distributions

belonged to either power-law or lognormal ones. Such distributions included Pareto, log-gamma, Frechet, log-Pearson, and lognormal. For all 24 species the distributions of attribute frequencies were skewed and leptokurtic (e.g., Figure 1). This means that, for each species, participants reported a large number of non-repeating attributes creating a long tail with a compact set of frequently agreed-upon attributes.

4 Discussion: Conceptual Modeling Is Obsolete. Long Live Conceptual Modeling

An emerging conceptual modeling challenge is modeling unpredictable and often unique user information from heterogeneous and distributed audiences. Addressing this challenge is difficult using traditional abstraction-driven modeling premised on *a priori* availability of “complete” specification of the kinds of data users would be contributing.

In this paper, we explored the possibility of omitting conceptual modeling and storing user input in flexible databases, such as an instance-based database. Based on the empirical evidence presented above, the instance-based approach with no conceptual modeling appears to meet the objectives of projects that engage distributed heterogeneous audiences better than the two class-based approaches (one based on intuitive and one based on useful classes). The diversity of attributes provided by non-experts makes it extremely difficult to *a priori* specify classes capable of capturing these attributes. For example, among reported attributes, some appear to be applicable to a particular instance (e.g., *deformed fin*), while some pertain to emotional evaluation of instances (e.g., *scary*). These kinds of attributes are likely to be unique to each situation and each person. Creating appropriate “containers” in advance (i.e., during conceptual modeling) to store these attributes is practically infeasible.

At the same time, the overall distribution of attributes resembles a long-tail with agreement on a core set of attributes and a large number of idiosyncratic ones. This suggests that attributes reflect some underlying regularities or shared perceptions of domain phenomena [40]. It is also notable that many attributes provided (here, 51.1%, see Table 3) by non-experts overlap with those established for species identification. At the same time, as seen from the categorical responses (see Table 1) participants fail to accurately classify at the species-level. This means that non-experts supply attributes that can be potentially used to identify instances at the species-level – a task shown to be mostly unattainable when classification is elicited directly.

Based on the evidence presented, there appears to be much value in avoiding traditional class-based conceptual modeling especially for IS aimed at managing distributed heterogeneous data. Does this spell the end of conceptual modeling in these settings? We argue that such a conclusion is premature, but making conceptual modeling relevant requires rethinking its role in IS development. Below we propose three promising approaches for future research to enhance value of conceptual modeling.

First, conceptual modeling can be used as a sensitizing tool rather than a formal specification that directly shapes IS design objects. For example, analysts can randomly sample potential users (e.g., potential citizen scientists) and ask to describe instances of interest (e.g., birds, cosmic bodies, material assets) using attributes. These attributes can then be analyzed to get an early glimpse into what actual data

may look like. This may reveal potential data conflicts and suggest ways to handle them. In this sense attributes become “thick descriptions” (as in ethnography or case research) that permit communication about issues in a domain with various stakeholders and guide design choices. This approach opens a new research stream aimed at developing, evaluating and improving models as sensitizing tools.

The second major direction for research deals with the issue of paradigmatic (e.g., ontological) assumptions that underlie IS development. While flexible database technologies appear well-suited for IS implementations, several issues arise, including: (1) how to design flexible data models and (2) how to choose an appropriate model for a given project. For instance, any flexible data model that stores data in a more or less structured manner (e.g., in terms of instances) adopts (implicitly or explicitly) ontological, epistemological, axiological and other paradigmatic assumptions about what reality is made of, what is valuable to capture, and how to best capture pertinent aspects of reality. For example, the instance-based data model follows philosophy of Mario Bunge and cognitive theories and assumes that reality is made of (unique) instances that possess properties [29]. Here we experimentally demonstrate that embedding these assumptions in IS leads to attainment of several desirable goals. The question arises, however, whether different paradigmatic assumptions are germane to different projects. For example, if IS resides in the context characterized by continuities rather than discrete instances [14], should analysts specify a flexible data model founded on these assumptions? Conceptual modeling research is engaged in rich and on-going discourse on these issues [14, 41-44]. Increased interest in distributed heterogeneous data motivates continued attention to paradigmatic assumptions in IS development.

Third, conceptual modeling research can begin addressing the issue of modeling under hybrid abstraction-based/no modeling paradigms. In practice most IS are likely to be on different points on the “no modeling” development continuum, as some aspects of a system could remain relatively fixed and amenable to abstraction-driven modeling. For example legal, security and reporting considerations could be embedded in software consistent with some fixed convention rather than left open to judgment of individual users. Similarly, a requirement to exchange data with legacy systems may suggest pre-specifying some structures in advance. This raises questions about how to integrate a no-modeling paradigm with traditional abstraction-driven modeling. Currently little is known about these issues and much scope exists in research on appropriate balance between different modeling approaches.

References

1. Fry, J.P., Sibley, E.H.: Evolution of Data-Base Management Systems. *ACM Computing Surveys* 8, 7–42 (1976)
2. Zuboff, S.: *In the age of the smart machine: The future of work and power*. Basic Books (1988)
3. Mylopoulos, J.: Conceptual Modeling and Telos. In: Loucopoulos, P., Zicari, R. (eds.) *Conceptual Modeling, Databases, and CASE: an Integrated View of Information Systems Development*, pp. 49–68. John Wiley & Sons, Inc., New York (1992)
4. Wand, Y., Weber, R.: Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research* 13, 363–376 (2002)
5. Hand, E.: People Power. *Nature* 466, 685–687 (2010)

6. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing Systems on the World-Wide Web. *Communications of the ACM* 54, 86–96 (2011)
7. March, S., Hevner, A., Ram, S.: Research Commentary: An Agenda for Information Technology Research in Heterogeneous and Distributed Environments. *Information Systems Research* 11, 327–341 (2000)
8. Brynjolfsson, E., Hu, Y.J., Simester, D.: Goodbye Pareto Principle, Hello Long Tail: The Effect of Search Costs on the Concentration of Product Sales. *Management Science* 57, 1373–1386 (2011)
9. Lyytinen, K., Yoo, Y.: Research Commentary: The Next Wave of Nomadic Computing. *Information Systems Research* 13, 377–388 (2002)
10. Krogstie, J., Lyytinen, K., Opdahl, A.L., Pernici, B., Siau, K., Smolander, K.: Mobile Information Systems - Research Challenges on the Conceptual and Logical Level. In: Olivé, À., Yoshikawa, M., Yu, E.S.K. (eds.) ER 2003. LNCS, vol. 2784, pp. 124–135. Springer, Heidelberg (2003)
11. Goodchild, M.: Citizens as Sensors: The World of Volunteered Geography. *GeoJournal* 69, 211–221 (2007)
12. Chen, P.: The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems* 1, 9–36 (1976)
13. Kaldor, N.: Capital Accumulation and Economic Growth. In: Lutz, F.A., Hague, D.C. (eds.) *The Theory of Capital*, pp. 177–222. Macmillan, London (1961)
14. Mylopoulos, J.: Information Modeling in the Time of the Revolution. *Information Systems* 23, 127–155 (1998)
15. Lukyanenko, R., Parsons, J.: Rethinking Data Quality as an Outcome of Conceptual Modeling Choices. In: 16th International Conference on Information Quality, pp. 1–16 (2011)
16. Parsons, J., Lukyanenko, R., Wiersma, Y.: Easier Citizen Science is Better. *Nature* 471, 37 (2011)
17. McGinnes, S.: Conceptual Modelling for Web Information Systems: What Semantics Can Be Shared? In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 4–13. Springer, Heidelberg (2011)
18. Ambler, S.: Agile database techniques: Effective strategies for the agile software developer. Wiley (2003)
19. Anwar, S., Parsons, J.: An Ontological Foundation for Agile Modeling with UML. In: Americas Conference on Information Systems (2010)
20. Chen, P.P.: Suggested Research Directions for a New Frontier – Active Conceptual Modeling. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 1–4. Springer, Heidelberg (2006)
21. Liu, C., Chrysanthis, P.K., Chang, S.: Database schema evolution through the specification and maintenance of changes on entities and relationships. In: Loucopoulos, P. (ed.) ER 1994. LNCS, vol. 881, pp. 132–151. Springer, Heidelberg (1994)
22. Ma, Z.M., Yan, L.: A Literature Overview of Fuzzy Database Modeling. *Journal of Information Science and Engineering* 24, 189–202 (2008)
23. Braun, S., Schmidt, A., Walter, A., Nagypal, G., Zacharias, V.: Ontology Maturing: A Collaborative Web 2.0 Approach to Ontology Engineering. In: 16th International World Wide Web Conference WWW 2007 (2007)
24. Robal, T., Haav, H.-M., Kalja, A.: Making web users' domain models explicit by applying ontologies. In: Hainaut, J.-L., et al. (eds.) ER Workshops 2007. LNCS, vol. 4802, pp. 170–179. Springer, Heidelberg (2007)

25. Lukyanenko, R., Parsons, J.: Unintended Consequences of Class-Based Ontological Commitment. In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 220–229. Springer, Heidelberg (2011)
26. Roussopoulos, N., Karagiannis, D.: Conceptual Modeling: Past, Present and the Continuum of the Future. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Mylopoulos Festschrift. LNCS, vol. 5600, pp. 139–152. Springer, Heidelberg (2009)
27. Potts, C.: Requirements Models in Context. In: IEEE International Symposium on Requirements Engineering, pp. 102–104 (1997)
28. Lukyanenko, R., Parsons, J.: Conceptual Modeling Principles for Crowdsourcing. In: 1st International Workshop on Multimodal Crowd Sensing, pp. 3–6 (2012)
29. Parsons, J., Wand, Y.: Emancipating Instances from the Tyranny of Classes in Information Modeling. ACM Transactions on Database Systems 25, 228–268 (2000)
30. Angles, R., Gutierrez, C.: Survey of Graph Database Models. ACM Computing Surveys 40, 1–39 (2008)
31. Abiteboul, S.: Querying Semi-Structured Data. In: Afrati, F.N., Kolaitis, P.G. (eds.) ICDT 1997. LNCS, vol. 1186, pp. 1–18. Springer, Heidelberg (1996)
32. Lukyanenko, R., Parsons, J.: Impact of Conceptual Modeling Approaches on Information Quality: Theory and Empirical Evidence. In: 17th International Conference on Information Quality, p. 5 (2012)
33. Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M., Boyesbraem, P.: Basic Objects in Natural Categories. Cognitive Psychology 8, 382–439 (1976)
34. Smith, L.B.: Emerging ideas about categories. In: Gershkoff-Stowe, L., Rakison, D.H. (eds.) Building Object Categories in Developmental Time, pp. 159–175. L. Erlbaum Associates, Mahwah (2005)
35. Bunge, M.: Treatise on basic philosophy: Ontology I: The furniture of the world. Reidel, Boston (1977)
36. Stokes, D.W., Stokes, L.Q., Lehman, P.E.: The stokes field guide to the birds of north america. Little, Brown, New York (2010)
37. Newcomb, L.: Newcomb's wildflower guide: An ingenious new key system for quick, positive field identification of the wildflowers, flowering shrubs and vines of northeastern and north central north america. Little, Brown and Company, New York (1977)
38. McClane, A.J.: McClane's field guide to freshwater fishes of north america. Holt Paperbacks, New York (1978)
39. Phillips, R.: Mushrooms & other fungi of north america. Firefly Books, Richmond Hill (2005)
40. Veres, C.: Concept modeling by the masses: Folksonomy structure and interoperability. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 325–338. Springer, Heidelberg (2006)
41. Wand, Y., Weber, R.: On Ontological Foundations of Conceptual Modeling: A Response to Wyssusek. Scandinavian Journal of Information Systems 18, 127–138 (2006)
42. Hirschheim, R., Klein, H.K., Lyytinen, K.: Information systems development and data modeling: Conceptual and philosophical foundations. Cambridge University Press, Cambridge (1995)
43. March, S., Allen, G.: Toward a Social Ontology for Conceptual Modeling. In: 11th Symposium on Research in Systems Analysis and Design, pp. 57–62 (2012)
44. Guarino, N., Guizzardi, G.: In the Defense of Ontological Foundations for Conceptual Modeling. Scandinavian Journal of Information Systems 18, 115–126 (2006)

Cognitive Mechanisms of Conceptual Modelling

How Do People Do It?

Ilona Wilmont¹, Sytse Hengeveld²,
Erik Barendsen¹, and Stijn Hoppenbrouwers^{1,3}

¹ Radboud University Nijmegen,
Institute for Computing and Information Sciences,
P.O. Box 9010, 6500 GL, Nijmegen, The Netherlands
{i.wilmont,e.barendsen}@cs.ru.nl

² Everest B.V.
Reitscheweg 55, 5232 BX, 's Hertogenbosch, The Netherlands
s.hengeveld@everest.nl

³ HAN University of Applied Sciences,
P.O. Box 2217, 6802 CE, Arnhem, The Netherlands
stijn.hoppenbrouwers@han.nl

Abstract. Conceptual modelling involves many higher order cognitive processes, such as relational reasoning and abstraction, which are based on integration and maintenance of information. Evidence from cognitive psychology suggests that these processes are subject to individual differences which cannot be explained by training and experience alone. In this review, we study how the cognitive processes that enable modelling interact to produce modelling behaviour, and where in this process we can find individual differences that may explain some of the variation in performance seen in actual modelling settings. We discuss interaction between working memory, executive control and attention as they facilitate relational reasoning and abstraction, which we consider to be key cognitive processes in modelling. Eventually, a thorough understanding of modelling cognition can help us to provide better cognitive support for modellers.

Keywords: Abstract reasoning, Collaborative modelling, Executive control, Working memory, Attention.

1 Introduction

Conceptual modelling is a core activity in system analysis [27], [53], involving reasoning with concepts and the relations between them. Good models rely on principles like modularity, abstraction and hiding [56]. Relevant information to the part of the system under study has to be related, and irrelevant information be made inaccessible so that it cannot exercise unwanted influences. However, there are great differences between modellers in their approach to forming abstractions. Thus, the quality of the resulting abstractions varies [67], sometimes to the detriment of the modelling session. In fact, “practitioners report that conceptual modelling is difficult and that it often falls into disuse within their organisations” [66].

Yet, modelling provides a framework for communication between developers and users, helps analysts understand a domain, provides input for the design process, and documents original requirements for future reference [66].

Hence, we need a more thorough understanding of abstraction in modelling so that targeted interventions for weak modellers may be provided. For this, we must identify which cognitive variables are involved and how they interact. In this article, we provide a high-level overview of these variables, which serves as a starting point for further, more fundamental study. We propose that relational reasoning and abstraction are key cognitive processes in modelling. They depend on the selection, maintenance and integration of relevant information, with constant monitoring for consistency and integrity. All these processes make use of executive control and goal pursuit, which in turn depends on working memory capacity [6], which is influenced by attention span [38] and emotional markers [8].

We begin with a brief review of relevant modelling concepts, then we continue with a discussion of what abstraction is and how it influences relational reasoning in modelling. Then, we relate this to the concept of executive control and its facilitators. Finally, we discuss sources of individual differences and possible implications for training modellers.

2 A Conceptual Analysis of Modelling

First of all, for clarification, we will briefly review modelling concepts. A *model* is an abstract unambiguous representation of a domain of interest, comprising concepts and relations [52], [22], which illustrates the behaviour and structure of a real-world system. In an ideal situation, the model should reflect the goals of the modellers and the stakeholders, and their knowledge and experience of the domain. Individuals involved in the modelling session are either modellers or model viewers. The *modeller* creates the model, first conceiving a mental model of the domain by reasoning from his prior experience and with the input he obtains from the environment [52], [27], [36]. His conception must then be written down as accurately as possible using a certain modelling language [27]. The *model viewer* is concerned with reading and understanding the model. He constructs an internal representation of the domain based on the information contained in the model, the specifics of the modelling language's grammar and symbols and his own previous experience [27]. Afterwards, his feedback provides input for the next iteration. Thus, *modelling* is the process of purposely creating a model based on the modeller's conceptions of the domain [52].

Modelling can be seen as a type of ill-structured problem solving. The initial state, the permissible operators, the optimal solution path and the goal state are not clearly defined or described [14]. For example, imagine designing a system which has to handle incoming data about how often employees have taken sick leave, in order to calculate their pension. Data has to be correct, complete and on time, so certain checks should be in place. In addition, all sorts of exceptions to the rule have to be built into the system. Decisions have to be made on

how to represent the flow of information, about which activities can be scoped within one flow and which ones form different flows. Such an ill-structured problem requires a cyclic approach. Information must be interpreted and structured. Abstraction helps the modeller to distinguish which properties are shared by certain activities, as well as to scope the information and to integrate various perspectives. At this point, the modeller can start thinking about a solution strategy, monitoring and evaluating each solution step as he proceeds [51]. Unfortunately, there is a large gap between what modellers *know* about a domain and how they make explicit their *representation* of it to others [28]. Moreover, the reasoning process is influenced by the amount of domain knowledge and the extent to which the forming of relations is required [36].

The discrepancy between the information known and information represented can be more explicitly described, respectively, in terms of the distinction between reading and writing a model [27]. Reading and writing are significantly different processes which will nevertheless alternate frequently within a modeller as he pauses to reflect, or listens to feedback. In the case of writing, the way the modeller understands the domain may not be accurately reflected in his model, while, in the case of reading, the understanding of the domain that the model viewer has constructed from the model may not match what the model is intended to represent. Active reasoning about the domain, if done by both modellers and model viewers, may eliminate these discrepancies [27]. Comprehension based on active reasoning leads to a deeper understanding of the domain, which is not the case with the passive comprehension based on recall. Active reasoning demands that viewers use the information represented to construct an answer to a problem related to the domain, an answer which cannot be immediately deduced from the model. This is true regardless of the modelling technique used [27].

The influence of domain knowledge on the reasoning process depends on the complexity of the relations involved. An expert in a domain should have his knowledge about it efficiently organised in his mind and know how to use the information. This would allow him to see meaningful patterns easily, and thus realise their strategic implications. However, solely domain knowledge is not always sufficient for solving a reasoning problem [39]. For this, *integration* of relations is required. Maier [39] shows that only once something “related to integration” was cued, solutions began to emerge. When relational integration fails, “inadequate responses based on native and acquired mechanisms are applied” [39]. This has been illustrated in a study on the effect of domain knowledge in modelling tasks. In tasks that require little relational integration, such as those in which extracted knowledge is directly represented in the model, domain knowledge has no effect on modelling performance. However, domain knowledge greatly improves performance on tasks which do require relational integration [36].

3 The Cognitive Mechanism of Modelling

In order to understand why difficulties in modelling occur so widely, we have to take the mechanism of modelling into account. As previously mentioned, comprehension is achieved through relational reasoning, making use of both experience

and new knowledge. Abstraction leads to a model conveying only the essential information relevant to a certain goal. It determines the way concepts are represented in the model. Also, by reaching a higher abstract understanding of a relation, reasoners are able to make inferences beyond the direct consequences [40]. Abstracting to the right level is strongly subject to individual differences, both in terms of experience and capacity, which will be discussed later. We will first discuss the forms and core properties of abstraction to help us understand how abstraction shapes relations. Then, we elaborate on relational reasoning in modelling.

3.1 Concepts and Relations

The main forms of abstraction we encounter during modelling are *concepts* and *relations*. Generally speaking, concepts are “what enables us to interpret situations in terms of previous situations that we judge as similar to the present” [25]. They participate in the generation of meaning based on a vast body of concrete knowledge. Without the information contained in a concrete situation, the concept would not be able to acquire any meaning [54]. Thus, the context dynamically influences the interpretation and structure of a concept [25]. For example, *fast* can only be truly appreciated when considered in relation to a set of observations of different speeds.

Formally, a relation is a “binding between a relation symbol and a set of ordered tuples of elements” [30], such as *faster(hare, tortoise)*. A relation describes the behaviour between two or more concepts. Relations are needed for constructing a process, in which they may acquire different complexities. Relational complexity is determined by the “number of related dimensions that need to be considered jointly to arrive at the correct solution” [19]. As is the case with concepts, a relation also does not become meaningful without a thorough, concrete understanding of the concepts it binds. Hence, concepts and relations depend on each other to acquire meaning. In a model, relations determine the position of different concepts. Relational knowledge has three essential properties which must be satisfied at all times during the reasoning process [30]. Representations have to maintain form across different levels of abstraction (*structure consistency*), and retain their meaning when participating in compound representations (*compositionality*). Also, if one can understand the meaning of a relation, then one can generate novel instances for that specific relation (*systematicity*). These properties ensure that relations retain their integrity, both internally and in the context of the whole model.

3.2 Properties of Abstraction

To begin with, an abstraction represents the innermost essence of a concept in a given context [3]. In an early theory of abstraction, George Berkeley (1685 - 1753) proposed that abstraction occurred through a “shift in attention”; one can focus on a particular feature of a single object, and let that feature represent a whole group of objects [9]. Rosch et al. [55] have identified a highly inclusive level of

abstraction that most people perceive as 'basic', for which Berkeley's view might apply. Instances at this level possess a great number of common attributes and have similar motor programs. As a consequence, potential category members may be identified by averaging the shapes of other members. However, a nuance to this view is necessary, as abstractions only become meaningful through coherent relations between features. Single distinctive attributes or random collections of properties turn out to be highly inadequate [3], [37].

A second property of abstraction is that it allows us to reason with generalities rather than instances. When observing a set of concrete instances, one can infer similar properties and use this knowledge in combination with experience to classify novel objects. As this process advances, concrete objects are replaced by simple propositions [21]. Abstract thought can then be used to perform mental operations on such object representations and possible actions in the mind [48]. It should be noted, though, that abstraction and generality are not synonymous. Rather, generality is required for abstraction, but abstraction is also required for generalisation [3]. This cyclic statement poses a problem: it implies that a certain abstract concept has to be present before any abstraction can be made. We can solve this by viewing concrete concepts as the aforementioned simple abstract representations [21] without breaking any rules of abstraction, for any mental representation is an abstraction and not the real object.

Thirdly, abstractions can form the basis for the creation of more complete descriptions [3]. This is illustrated in a study of how linguistic ideas are abstracted [11]. Upon hearing and later recalling semantically related sentences, the abstracted representations ended up being far more complete than the original sentences ever were, because relations were supplemented with available knowledge. Hence, we see how the link to an individual's experience is made through active memory (re)construction and how experience influences abstract ideas.

A fourth feature is the relativity of abstraction. Barsalou [7] writes that abstractions are dynamic, tailored to a purpose, temporary, flexible, become more easy to use with practice, and involve attention shifting. Formation of abstractions are influenced by available knowledge structures, intentions, goals, experience, and the beholder's context. Therefore, people differ in what they consider abstract or concrete, and this may change as experience develops and contexts change. In practice, people reason on a certain 'preferred level of abstraction' [26], which appears to be mostly context-dependent, fluctuates widely over time and is independent of the capacity to abstract and of general intelligence. In some cases, the automaticity of previous experiences and habits primes future abstractions regardless of the context, even in the presence of accessible abstract rules.

Finally, abstractions are complex, and the required mental effort to form abstractions increases proportionally with the level of abstraction, which is independent of task difficulty [15]. The balance between the level of detail and the generic structure in a model is a delicate one. If we move down a level towards describing concrete instances, knowledge becomes much more specialised. Details become voluminous and minutely distinguishable as they may be semantically

or visually related. Considering them all would lead to an attention overload. On the other hand, if detailed information is lacking, grasping the meaning of an abstraction becomes impossible. Before being able to see high-level structures and generalities which can be used in productive thinking, a whole range of behaviours a concept may exhibit has to be well understood [3].

3.3 Relational Reasoning in Modelling

Relational reasoning is “the ability to consider relationships between multiple mental representations” [19], and is implied in tasks requiring conceptual relations [12]. The purpose of reasoning is to form conclusions, judgments, or inferences from facts or premises. The resulting relations eventually make up the model. Reasoning combines separate past experiences and novel input in such a manner as to achieve a goal, or to meet the demands of a situation. In this context, it is important to distinguish between two types of information processing that may take place during reasoning: analytic and nonanalytic cognition [33]. Analytic cognition involves breaking a stimulus into relevant and irrelevant features, and generality stems from a certain configuration of recombinable units. In contrast, nonanalytic cognition relies on referring back to specific prior episodes, thereby incorporating real-world knowledge, and generality results from finding analogies between similar situations [33]. Both involve abstracting relevant features, and maintaining and comparing them in mind. In the latter case, though, recombination has already taken place.

Abstraction in relational reasoning manifests as flexible switching between concrete and abstract representations, and shifts in focus on certain relevant properties. These could be considered meta-relations across knowledge. Bearing the properties of relational knowledge in mind (structure consistency, systematicity and compositionality [30]), we can distinguish two types of switching strategies. The first type of switching could be considered ‘vertical’: shifting up and down between abstraction levels through generalisation and instantiation [61]. This is how the model is made *meaningful* by actively connecting generic concepts to what the evolving model means in terms of the activities and objects a modeller or user encounters in his daily environment. Sometimes the connection between concrete and abstract goes awry. Usually, people know only partly how the complex systems they work with function, rendering it very hard to make explicit procedural or tacit knowledge [43]. Such concrete knowledge gaps can result in faulty conceptions of a domain, a failure to monitor understanding and progress, and thereby hamper success at problem solving. It has been observed, though, that if concepts turn out to be too abstract to understand, unconscious reduction of the abstraction level takes place [31]. The second type of switching is ‘horizontal’, in the sense that even though attention shifts to a different focus or context, the same concept is still being considered. If the context shifts, so do the features that have to be considered and they may acquire totally new meanings [39]. In such cases structure consistency and compositionality must be monitored to maintain model integrity. Thus, relations can be made between concepts based on those properties relevant for the situation, otherwise they may

be ignored [56], [61]. However, omitted information is only made inaccessible so that it does not affect the part of the system under consideration. This is known as *hiding* [56].

4 Reasoning Explained in Cognitive Processes

In order to avoid violating any of the properties of relational knowledge during reasoning so that meaningful relations may be formed, through analytic and non-analytic cognition, involvement of executive control functions is required [16], [19]. Essentially, executive functions control and coordinate behaviour so that people can achieve goals in an efficient manner. It allows people to inhibit distractions and irrelevant routine behaviours, and monitor their progress through controlled attention, regardless of the specifics of the task [29]. Also, it facilitates critical reflection, which is essential to consolidating knowledge and learning [65]. When central executive functioning is disrupted, performance on conditional reasoning tasks suffers considerably [63]. In an extensive review, Alvarez & Emory [2] find that the processes underlying executive functions are inhibition and switching, working memory and selective and sustained attention. We will discuss modelling in terms of each of the underlying processes.

4.1 Executive Control and Goal Pursuit: Facilitators of Reasoning

During modelling, a modeller may participate in diverse cognitive activities, such as to select the relevant information, and to regulate and monitor his selection in case of multiple simultaneous inputs. He must read, interpret and comprehend this information, and match his own mental representation with what other modellers are saying and writing. Moreover, modellers should not only monitor themselves, but also monitor others in the flow of discussion so that they can react appropriately to other participants, providing them with information to interpret [67]. At the end of the session, the modeller needs to relate the modelling goals and the users' needs to the model created to ensure final model quality [57]. Executive control is thus not only used to monitor the information being processed, but also to interact closely with goal pursuit processes to provide the modeller with a direction in which to operate. Through response inhibition and performance monitoring, the modeller modulates how he responds to a stimulus depending on his goal and context, allowing him to adapt to or to produce consistent output in a novel situation [10]. Goal-seeking behaviour depends very strongly on goal maintenance. Goal-directed selection of information is essential in order to deal with the structure imposed by the reasoning task [16]. It enables searching through memory and selecting relevant information to substitute into the task structure. Also, goals associated with affective markers determine the amount of effort invested in achieving the goal in question [8], and the likelihood of engaging executive control [24]. When positive affect becomes associated or coactivated with a goal concept, motivation and readiness to achieve that

particular goal is facilitated, while negative affect is assumed to put preexisting goals on hold [1]. Using affective signals, qualitatively different alternatives can be non-consciously evaluated.

4.2 Working Memory: The Driving Force

Facilitation of executive control depends on the working memory (WM) system; in fact, it is part of the WM system, conceptualised as the ‘central executive’ component in the model of WM by Baddeley et al. [5]. WM is a limited capacity memory system that deals with both the storage and manipulation of information. Information in WM, whether newly perceived or retrieved from long-term memory (LTM) is ephemeral by nature, but can be maintained by repetition. Evidence suggests that WM is involved in many complex cognitive processes relevant to modelling, such as (relational) reasoning [19], comprehension [38], abstract thought [15], unconscious goal pursuit [23] and problem solving [38]. Across these domains, the essential functional role of WM is primarily concerned with activation, selection, maintenance and manipulation of information to make it relevant for goal and context [44], [46]. WM is suggested to be the ‘workspace’ where relations are constructed and altered [30]. The WM system maintains representations on a certain level of abstraction [15].

In the Baddeley model [5], storage is composed of an episodic buffer, aided by a phonological loop for processing auditory and speech-based information, and a visuospatial sketchpad for visual images in the context in which we perceive them in the everyday world. Control and manipulation of information happens through the central executive. A slightly different focus is taken by Cowan [17]. He proposed the *Embedded Process Theory*, a hierarchically organised theory in which working memory is based on activated information which is “unusually accessible”. Within the LTM store, there is a subset of activated information. This is the short-term store. Within this short-term store there is yet another subset, which is within the focus of attention. The focus of attention determines what information is available to WM, thereby exercising control over WM, while attention span capacity determines how much information can be activated at any given time. Information within the focus of attention is typically used for tasks requiring strong executive control, whereas information within the short-term store can be used for automated, familiar tasks. Despite subtle differences in conceptualisation, the essence of the WM models is that WM is a control mechanism operating on information, residing in LTM stores, made easily accessible.

There is evidence to suggest that for the most part, WM is a fundamental, domain general processing capacity involved in both elementary and complex cognitive processing [38]. However, there does appear to be a certain degree of WM specialisation for specific types of information on a deeper level. Ample evidence points out a differentiation between spatial and non-spatial WM [20], [46]. Spatial WM resources are strongly linked to performance on abstraction and executive control tasks [47]. Concrete reasoning, in contrast, appeared to draw much more on non-spatial, visual WM resources [34]. A constraint on the system is the severely limited number of items that can be held in WM for immediate

use. Chen & Cowan [13] show that WM is generally constrained to 3-5 chunks of meaningful items of information, if rehearsal is actively prevented. There is no question that active interference of competing items of information plays a major role in the quick decay of WM content, but the extent to which time-based decay of activated information traces plays a role is not clear. However, in a realistic situation like modelling, it seems unlikely that there is time for temporal decay due to the high volume of interaction and discourse taking place. Thus, we are most likely dealing with interference most of the time. This only emphasises the importance of maintaining a focus during modelling.

4.3 Attention: The Capacity Mediator

Attention allows one to concentrate on a certain set of objects or tasks, in favour of others, thereby assigning a bias to internal representations which serves subsequent cognitive processing. It translates goals into behaviour by orienting an individual towards goal-relevant information, using the affective signals mentioned before to determine relevance [8]. Attention also supervises this process through executive control. Information entering the limited capacity of WM (WMC) is mitigated by attention [38] and time [41]. Both are critical resources to allow invocation of executive processes so that controlled processing can take place. If time is too short for executive processes to be activated, attentional resources cannot be allocated to invoke strategy use. As a consequence, performance on reasoning tasks relies more strongly on basic WM span [41]. Attentional resources are likely shared between different aspects of WM, as illustrated by Just et al. [35], who show that two WM tasks, which draw on different neural systems, show impaired performance when performed simultaneously. WM appears to recruit attention in the service of memory rehearsal [4]. In support of this, it is found that loading WM decreases control of attention over inhibitory and switching functions [32]. Hence, WMC relates directly to attentional control over executive functions. Furthermore, the relation between attention and WMC is influenced by the need for coordination of multiple knowledge representations, when people are required to attend to multiple representations simultaneously to determine which one to use [58]. If no coordination is required, WMC only determines how well stimulus-response associations are learned. If coordination is required, then WMC not only predicts learning performance but also how well people can shift between response strategies.

4.4 Sources of Individual Differences

Both WMC [42], [16] and attention span [26], [38] have proven to be significant sources of individual differences in abstraction, reasoning and problem solving. Even though the WM capacity limit is fixed within the 3-5 range, there are significant individual differences to be observed. Explanations for this are sought in storage and processing capacity variations. There is a debate as to whether the efficiency of processing ability is solely responsible for all WM functionality [42], or whether storage and processing capacities are independent and both contribute to overall individual differences [18]. In favour of the first perspective,

individuals with a high WMC were found to be much better at attending to relevant information and thus inhibiting irrelevant information. Low capacity individuals, in contrast, processed both target and distracter information, ending up with far less meaningful information overall and so rendering distractions detrimental to performance [16], [42]. This effect has been demonstrated for the auditory [39] and the visual modality [64]. On the other hand, in favour of physical storage capacity, Todd & Marois [62] show that in the absence of processing requirements, brain activity in the posterior parietal areas correlated with WM performance.

5 General Discussion

In modelling literature, many authors have made reference to either executive control or working memory in attempts to understand and support modelling. For instance, Sutcliffe & Maiden [60] implicitly identify differences in executive control processes in weak and strong novices. Pinggera et al. [49] relate the use of structure during modelling to higher quality models. Finally, Moody [45] emphasizes the role of WM in understandability of models. However, the psychological frameworks used do not describe modelling behaviour in terms of its facilitating variables. Furthermore, when designing protocols for modelling support, this only led weak modellers to rigidly follow the protocol and it did not impart understanding of the modelling problem [60]. Similarly, even though both WM capacity and attention span have been subject to training, it is unclear whether WMC can actually be increased. While WM training appears to improve task performance, Shipstead et al. [59] point out that it has not yet been shown that this effect is directly due to an increase in actual WM capacity. Since the mechanisms of WM transfer to other skill domains are unclear, the observed improvement might well be due to a learning effect. Also, transfer to improved attention could not be attributed to WM. Therefore, we need to study modelling beyond the behavioural level, going as far as to study its neural substrates. The neural workings can eventually be used as a model for targeted intervention.

To a certain extent, the strategy of creating awareness has yielded some results. Increasing attention to relevant features improves performance in abstract tasks [50]. Awareness of attending to features increased analytic processing. Explanation of this effect in terms of mental model theory [34] would state that mental models are being further elaborated by paying attention to those features that were still missing.

In conclusion, we argue for a cognitive system enabling modelling in which executive control monitors information processing for integrity of representations. Switching between abstraction levels and different contexts, and inhibiting distracting information are primary processes. Also, it interacts with goal pursuit to maintain relevant goals, making use of affective body signals. Executive control is part of WM, which facilitates storage of representations and processing of content information. WM has a limited capacity, which is subject to individual differences. A source of differences is the ability to mediate attention between relevant and irrelevant stimuli competing for processing.

References

1. Aarts, H., Custers, R., Holland, R.W.: The nonconscious cessation of goal pursuit: When goals and negative affect are coactivated. *Journal of Personality and Social Psychology* 92(2), 165–178 (2007)
2. Alvarez, J.A., Emory, E.: Executive function and the frontal lobes: A meta-analytic review. *Neuropsychology Review* 16(1), 17–42 (2006)
3. Arnheim, R.: Visual Thinking. University of California Press (1969)
4. Awh, E., Jonides, J.: Overlapping mechanisms of attention and spatial working memory. *Trends in Cognitive Sciences* 5(3), 119–126 (2001)
5. Baddeley, A., Logie, R., Bressi, S., Sala, S.D., Spinnler, H.: Dementia and working memory. *The Quarterly Journal of Experimental Psychology Section A* 38(4), 603–618 (1986)
6. Bailey, H., Dunlosky, J., Kane, M.: Why does working memory span predict complex cognition? testing the strategy affordance hypothesis. *Memory & Cognition* 36, 1383–1390 (2008), doi:10.3758/MC.36.8.1383
7. Barsalou, L.W.: Abstraction in perceptual symbol systems. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 358(1435), 1177–1187 (2003)
8. Bechara, A., Damasio, H., Damasio, A.R.: Emotion, decision making and the orbitofrontal cortex. *Cerebral Cortex* 10(3), 295–307 (2000)
9. Berkeley, G., Krauth, C.P.: A Treatise Concerning the Principles of Human Knowledge. JB Lippincott & Co. (1878)
10. Berkman, E.T., Falk, E.B., Lieberman, M.D.: Interactive effects of three core goal pursuit processes on brain control systems: Goal maintenance, performance monitoring, and response inhibition. *PLoS ONE* 7(6), e40334 (2012)
11. Bransford, J.D., Franks, J.J.: The abstraction of linguistic ideas. *Cognitive Psychology* 2(4), 331–350 (1971)
12. Cattell, R. (ed.): Intelligence: Its Structure, Growth and Action. Advances in Psychology, vol. 35. North Holland (1987)
13. Chen, Z., Cowan, N.: Core verbal working-memory capacity: The limit in words retained without covert articulation. *The Quarterly Journal of Experimental Psychology* 62(7), 1420–1429 (2009)
14. Chi, M.T.H., Glaser, R.: Problem solving ability. In: Sternberg, R.J. (ed.) *Human Abilities: An Information Processing Approach*, ch. 10. Freeman, New York (1985)
15. Christoff, K., Keramati, K., Gordon, A., Smith, R., Mädler, B.: Prefrontal organization of cognitive control according to levels of abstraction. *Brain Research* 1286, 94–105 (2009)
16. Chuderska, A.: Executive control in analogical mapping: Two facets. In: Proceedings of the 32nd Annual Conference of the Cognitive Science Society, pp. 2749–2754 (2010)
17. Cowan, N.: An embedded-processes model of working memory. In: Miyake, A., Shah, P. (eds.) *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, ch. 3, pp. 62–101. Cambridge University Press (1999)
18. Cowan, N., Saults, J.S., Morey, C.C.: Development of working memory for verbal-spatial associations. *Journal of Memory and Language* 55(2), 274–289 (2006)
19. Crone, E.A., Wendelken, C., Van Leijenhorst, L., Honomichl, R.D., Christoff, K., Bunge, S.A.: Neurocognitive development of relational reasoning. *Developmental Science* 12(1), 55–66 (2009)

20. Daneman, M., Tardif, T.: Working memory and reading skill re-examined. In: Coltheart, M. (ed.) *Attention and Performance*, vol. 12, pp. 491–508. LEA, Hillsdale (1987)
21. Diamond, A.: Bootstrapping conceptual deduction using physical connection: re-thinking frontal cortex. *Trends in Cognitive Sciences* 10(5), 212–218 (2006)
22. Dietz, J.: *Enterprise Ontology: Theory and Methodology*. Springer (2006)
23. Dijksterhuis, A., Aarts, H.: Goals, attention, and (un)consciousness. *Annual Review of Psychology* 61, 467–490 (2010)
24. Dixon, M.L., Christoff, K.: The decision to engage cognitive control is driven by expected reward-value: Neural and behavioral evidence. *PLoS ONE* 7(12), e51637 (2012)
25. Gabora, L., Rosch, E., Aerts, D.: Toward an ecological theory of concepts. *Ecological Psychology* 20(1), 84–116 (2008)
26. Gardner, R.W., Schoen, R.A.: Differentiation and abstraction in concept formation. *Psychological Monographs: General and Applied* 76(41), 1–21 (1962)
27. Gemino, A., Wand, Y.: Evaluating modeling techniques based on models of learning. *Communications of the ACM* 46(10), 79–84 (2003)
28. Gemino, A., Wand, Y.: A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering* 9(4), 248–260 (2004)
29. Gilbert, S.J., Burgess, P.W., et al.: Executive function. *Current Biology* 18(3), 110–114 (2008)
30. Halford, G.S., Wilson, W.H., Phillips, S.: Relational knowledge: The foundation of higher cognition. *Trends in Cognitive Sciences* 14(11), 497–505 (2010)
31. Hazzan, O.: Reflections on teaching abstraction and other soft ideas. *ACM SIGCSE Bulletin* 40(2), 40–43 (2008)
32. Hester, R., Garavan, H.: Working memory and executive function: The influence of content and load on the control of attention. *Memory & Cognition* 33(2), 221–233 (2005)
33. Jacoby, L.L., Brooks, L.R.: Nonanalytic cognition: Memory, perception, and concept learning. In: Bower, G.H. (ed.) *Psychology of Learning and Motivation*, vol. 18, pp. 1–47. Academic Press (1984)
34. Johnson-Laird, P.N.: Deductive reasoning ability. In: Sternberg, R.J. (ed.) *Human Abilities: An Information Processing Approach*, W.H. Freeman (1985)
35. Just, M.A., Carpenter, P.A., Keller, T.A., Emery, L., Zajac, H., Thulborn, K.R.: Interdependence of nonoverlapping cortical systems in dual cognitive tasks. *NeuroImage* 14, 417–426 (2001)
36. Khatri, V., Vessey, I., Ramesh, V., Clay, P., Park, S.: Understanding conceptual schemas: Exploring the role of application and is domain knowledge. *Information Systems Research* 17(1), 81 (2006)
37. Lakoff, G.: Cognitive models and prototype theory. In: Margolis, E., Laurence, S. (eds.) *Concepts: Core Readings*, ch. 18, pp. 391–421. The MIT Press (1999)
38. Lépine, R., Parrouillet, P., Camos, V.: What makes working memory spans so predictive of high-level cognition? *Psychonomic Bulletin & Review* 12, 165–170 (2005), doi:10.3758/BF03196363
39. Maier, N.R.: Reasoning in rats and human beings. *Psychological Review* 44(5), 365–378 (1937)
40. Manktelow, K., Fairley, N.: Superordinate principles in reasoning with causal and deontic conditionals. *Thinking & Reasoning* 6(1), 41–65 (2000)
41. McCabe, D.P.: The influence of complex working memory span task administration methods on prediction of higher level cognition and metacognitive control of response times. *Memory & Cognition* 38(7), 868–882 (2010)

42. McCollough, A., Vogel, E.: Your inner spam filter. *Scientific American Mind* 19(3), 74–77 (2008)
43. McDermott, R.: Why information technology inspired but cannot deliver knowledge management. *California Management Review* 41(4), 103–117 (1999)
44. Miyake, A., Friedman, N., Emerson, M., Witzki, A., Howerter, A., Wager, T.D.: The unity and diversity of executive functions and their contributions to complex frontal lobe tasks: A latent variable analysis. *Cognitive Psychology* 41(1), 49–100 (2000)
45. Moody, D.L.: The “physics” of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering* 35(6), 756–779 (2009)
46. Oberauer, K., Süß, H.M., Schulze, R., Wilhelm, O., Wittmann, W.: Working memory capacity - facets of a cognitive ability construct. *Personality and Individual Differences* 29(6), 1017–1045 (2000)
47. Owen, A.M., Downes, J.J., Sahakian, B.J., Polkey, C.E., Robbins, T.W.: Planning and spatial working memory following frontal lobe lesions in man. *Neuropsychologia* 28(10), 1021–1034 (1990)
48. Piaget, J.: *Zes Psychologische Studies*. Van Loghum Slaterus (1969)
49. Pinggera, J., Soffer, P., Zugal, S., Weber, B., Weidlich, M., Fahland, D., Reijers, H.A., Mendling, J.: Modeling styles in business process modeling. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) *BPMDS 2012 and EMMSAD 2012. LNBIP*, vol. 113, pp. 151–166. Springer, Heidelberg (2012)
50. Platt, R.D., Griggs, R.A.: Facilitation in the abstract selection task: The effects of attentional and instructional factors. *The Quarterly Journal of Experimental Psychology Section A* 46(4), 591–613 (1993)
51. Pretz, J.E., Naples, A.J., Sternberg, R.J.: Recognizing, defining and representing problems. In: Davidson, J.E., Sternberg, R.J. (eds.) *The Psychology of Problem Solving*. Cambridge University Press (2003)
52. Proper, H.A., Van Bommel, P., Hoppenbrouwers, S.J.B.A., Van der Weide, T.P.: A fundamental view on the act of modeling. In: Kizza, J., Aisbett, J., Vince, A., Wanyama, T. (eds.) *Advances in Systems Modeling and ICT Applications, Special Topics in Computing and ICT Research*, vol. 2, pp. 97–112. Fountain Publishers, Kampala (2006)
53. Renger, M., Kolfschoten, G., De Vreede, G.: Challenges in collaborative modelling: A literature review and research agenda. *International Journal of Simulation and Process Modelling* 4(3), 248–263 (2008)
54. Rosch, E.: Reclaiming concepts. *Journal of Consciousness Studies* 6(11-12), 61–77 (1999)
55. Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M., Boyes-Braem, P.: Basic objects in natural categories. *Cognitive Psychology* 8(3), 382–439 (1976)
56. Ross, D., Goodenough, J., Irvine, C.A.: Software engineering: Process, principles, and goals. *Computer* 8(5), 17–27 (1975)
57. Sedera, W., Rosemann, M., Gable, G.: Measuring process modelling success. In: *Proceedings of ECIS 2002* (2002)
58. Sewell, D., Lewandowsky, S.: Attention and working memory capacity: Insights from blocking, highlighting, and knowledge restructuring. *Journal of Experimental Psychology: General* 141(3), 444–469 (2012)
59. Shipstead, Z., Redick, T.S., Engle, R.W.: Is working memory training effective? *Psychological Bulletin* 138(4), 628–654 (2012)

60. Sutcliffe, A., Maiden, N.: Analysing the novice analyst: cognitive models in software engineering. *International Journal of Man-Machine Studies* 36(5), 719–740 (1992)
61. Theodorakis, M., Analyti, A., Constantopoulos, P., Spyros, N.: Contextualization as an abstraction mechanism for conceptual modelling. In: Akoka, J., Bouzeghoub, M., Comyn-Wattiau, I., Métais, E. (eds.) ER 1999. LNCS, vol. 1728, pp. 475–490. Springer, Heidelberg (1999)
62. Todd, J.J., Marois, R.: Posterior parietal cortex activity predicts individual differences in visual short-term memory capacity. *Cognitive, Affective, & Behavioral Neuroscience* 5, 144–155 (2005)
63. Toms, M., Morris, N., Ward, D.: Working memory and conditional reasoning. *The Quarterly Journal of Experimental Psychology Section A: Human Experimental Psychology* 46(4), 679–699 (1993, 2013)
64. Vogel, E.K., McCollough, A.W., Machizawa, M.G.: Neural measures reveal individual differences in controlling access to working memory. *Nature* 438(7067), 500–503 (2005)
65. Vygotsky, L.: *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press (1978)
66. Wand, Y., Weber, R.: Research commentary: Information systems and conceptual modeling - a research agenda. *Information Systems Research* 13(4), 363–376 (2002)
67. Wilmont, I., Barendsen, E., Hoppenbrouwers, S.J.B.A., Hengeveld, S.: Abstract reasoning in collaborative modeling. In: HICSS Proceedings, vol. 45 (2012)

A Semantic Analysis of Shared References

Roland Kaschek

The University of the Faroe Islands
rolandkaschek@gmail.com

Abstract. Models, according to common understanding, facilitate to make and share references. A reference is a deictic activity whose point of origin differs from its target. The activity of modeling rests on taking a reference's point of origin as a substitute for its target. In this paper I discuss two aspects of modeling, namely abstraction and direction-of-fit. I am going to show that representation, rather than abstraction, is key to modeling and that the direction-of-fit is not inherent to a model.

1 Introduction

A sign is a thing that stands for something else, its referent. Following Morris [15] one considers processes of sign usage, according to the dimensions syntax, semantics and pragmatics. Conceptual models are widely regarded as composite signs. The mentioned dimensions thus are landmarks in the discussion of conceptual modeling. To arrive at a definition of conceptual model I blend the views of Thalheim [21],[22], Frigg & Hartmann [6] and Kralemann & Lattmann [12]. By conceptual model I mean the class of those composite signs, that serve the members of a community equally well in achieving a given purpose. I call each such sign a representation. Typically conceptual models are used to create computer applications that aid the community members in practicing dedicated behaviors. This distinguishes conceptual models from signs in general, as the latter are not supposed to aid in practicing behaviors. However, Seidewitz [18], require a model to consist of utterances, that are either true or false. I suppose that conceptual models are models in the more general sense and in particular in the sense of Stachowiak [19]. According to [6] in the philosophy of science models increasingly are not considered as signs anymore.

The quotations in [16] indicate that several researchers in conceptual modeling more or less accept Stachowiak's [19] key characteristics of models: mapping characteristic (each model is a model of something, its so-called original or target), truncation characteristic (each model is devoid of a number of the characteristics of its original) and pragmatic characteristic (each model is supposed to aid a community in meeting a specific purpose). Further such characteristics have been listed by Thalheim [21]. In this paper I suppose that the term "mapping characteristic" means nothing but the modeling community's members' ability to share a reference, i.e. to refer coherently from a model to its original. By considering what the term mapping characteristic means I am applying a semantic angle to conceptual modeling. Harel & Rumpe [8] do that

too. Their focus, however, is on the meaning of modeling language constructs while I focus on the meaning of the model's original. I sketch a related theory and use it to show that in conceptual modeling representation takes precedence over abstraction and that being descriptive or prescriptive is not inherent to a model but rather is a characteristic of a reference carried out by a modeler. Due to space limitations as examples I use models in general rather than conceptual models. This is not an essential restriction.

Additionally to semantics further philosophical disciplines such as aesthetics, epistemology, ethics and ontology may be used to fertilize discourses. A lot of work on conceptual modeling is inspired by ontology. In particular the so-called Bunge-Wand-Weber ontology has played a major role. Ontology deliberates about the kinds of things that exist. The outcome of such deliberations is quite independent from what concerns me in this paper. I am thus going to ignore all that work. A number of related references, however, can be found via section 4 of [23].

2 A Brief Semantics of Shared References

For a conceptual model of a modeling community to have the mapping characteristic it is required that the members of that community somehow grasp the intended model original. Since the conceptual models in general are composite that grasping of the original can be understood as a complex process in which at first the elementary model parts are grasped and then are fit together according to the related model's instructions. To mentally construct the model original, however, is not all. That construction, by each community member, for proper common discourse, has to be taken as common point of reference. Moreover, the related utterances of fellow community members have to be interpreted as if these would have done the same.

The representations in a conceptual model include signs. Some of these may be independent and others may have signs as parameter. The most elementary activity in the mental construction of the original of a shared model thus is a deictic reference from a simple sign to its referent. Deictic references that start in similar signs and end up in similar things can be grouped together to form deictic reference schemas (DRS). A DRS thus enables coherent references from sets of similar signs to sets of similar things. Obvious quality aspects of DRS are: readiness (any community member, at will, can instantiate the DRS), invariance (for each community member the referent only marginally, if at all, depends on the individual schema instantiation) and coherence (the referent only marginally, if at all, depends on the community member instantiating the schema). I call a DRS successful if it has these three quality characteristics. Modeling projects, that do not arrive at successful DRS, obviously are going to be troubled. I reuse the concept of DRS from [9].

Modelers, who employ a modeling language, like the entity-relationship model (ERM) [3], to create a conceptual model work out three things: the model's thesaurus, its graph and its interpretation. The thesaurus (aka data dictionary

or data ontology) is a list of definitions of terms conventionally used in proper discourse. The graph (aka diagram) is a schema of the cohesions among the defined terms in the thesaurus. The interpretation is a mapping of the graph's elements to the terms defined in the thesaurus. It is thus chiefly by means of the thesaurus that modelers carry out the references required for benefitting from a model. It seems that the thesaurus' passive role as information repository in part is reflected in current work such as in [4] or [20]. Others, however, such as [16], ignore it entirely. The research direction indicated here seems to fit best the section 6 of Wand's & Weber's research commentary [23]. That paper, however, does not address the thesaurus explicitly. More research on how to obtain, use and maintain a thesaurus might aid in boosting the quality of conceptual models.

Carnap [2, p. 11] has emphasized the importance of making references for the purpose of modeling. The basic kinds of reference, as applying to conceptual modeling, seem to be: to refer to a thing-set in some universe of discourse (UoD); to characterize, by its context, a thing in such a thing set and to relate to each other things in that UoD. Carnap (pp.17) discusses, as an example, how the European railway system of his time could be modeled as a graph. He approaches his modeling problem in a way that is consistent with a binary ERM with 1 : 1 relationships and one-element entity sets only. The utility of models in general is not accidental. It rather can be attributed to the application of appropriate model creation and representation rules. Lockemann and Mayr [13] have denoted them by "modeling concept" and "representation concept", respectively. Additionally to using such proper rule sets can entirely account for a model's utility. In particular no similarity between a model and its original needs to be presupposed to explain its utility.

Marx [14] has noted that the: "... question whether objective truth can be attributed to human thinking is not a question of theory but is a practical question. Man must prove the truth i.e. the reality and power, the this-sidedness of his thinking in practice..." I take this as the identification of two dimensions of human thought. I denote them by constitution and deduction, respectively. Modeling communities capture and exploit knowledge about their world. To that end they explore and use DRS. Modeling to them becomes an experimental pursuit. It involves conducting experiments into how to figure out how to satisfactorily characterize, refer to and relate to each other things in some UoD. A model in consequence is a tool of a community. A successful model, in particular, is therefore tailored towards that community and depends on it. I find the idea that a successful tool is a negative image of its users in [5]. The empirical knowledge captured via a model is the knowledge that proceeding, such and such way, with such and such conventions under such and such circumstances one can, at acceptable cost and in given circumstances, successfully practice a dedicated behavior. That behavior often, but not necessarily, is about solving problems.

The graph of a conceptual model is a mutual formal definition of terms. It is thus an axiom system. It allows for many different interpretations, none of which is inherent to it. The practical importance of the so-called modeling grammar (the syntax of the modeling language used) and its usage rules cannot

be overrated. However, modelers in practice seem to act more on the meaning of the sign referents rather than on the structure of the model graph.

3 Modeling vs. Abstracting

It has been observed that a close relationship exists between abstraction and modeling. Booch et al., for example, say [1]: “What, then, is a model? Simply put, A model is a simplification of reality.” Similar utterances are quoted in [16, table 1]. Those views imply that modeling essentially is abstracting. Before I provide my counter-arguments I note that I have argued before against equating abstracting and modeling [10]. I also reuse an example [11] to show that there is a reason to consider things differently than Booch et al. do. Consider a conventional pocket-city-map of Torshavn, the capital of the Faroe Islands. Then most likely everyone agrees that it can be considered as a model of Torshavn. And of course it also represents an abstraction of Torshavn as many of that city’s characteristics are not reflected in that map. It does, for example, not tell how the wind feels when you walk the harbor in winter; how folks look like who walk the streets; how the music sounds in the pub during a jazz-concert and many things more. Obviously for the purpose of navigating through the city, in most cases, the ignored information is irrelevant. It would, nevertheless be false to simply consider that map as a mere abstraction of that city. In fact important usage characteristics of that map result from characteristics it does not inherit from its original: It has a legend, complies with map-related conventions, can be folded, be put into one’s pocket and you can have it on you when you walk the streets in order to get to some particular location. None of these characteristics is a characteristic of the city Torshavn.

The example suggests that models in general have characteristics that are not derived from their original. This point has already been made by Stachowiak and is also made in [21]. These authors seem, however, not, like I do, to attribute the specific utility of a model to its surplus characteristics. My first argument against Booch et al.’s view is that neither the abstraction nor the model is immediately accessible to the model users. Rather the model users initially only have access to a representation. From it they have to recover the intended abstraction. It is clear that in general for any abstraction there are many ways to represent it. From the set of usable representations one needs to be chosen that works best for the modeling community. Second, if a model would be just an abstraction then it would have characteristics of its original only. Therefore one could do with the model exactly what one could do with that abstraction. That would render the model superfluous since all that could be done with it could already be done with that abstraction and hence the model’s original. Third, a suitable abstraction might not to be at hand right away. It might, rather, have to be created. Often for that, as is well-known, one would get hold of an preliminary representation first. One then, by modifying the representation, would create a suitable abstraction. Fourth, often an instance of a model, just as Booch et al. do above, is falsely regarded as “the” model. Since a model is the equivalence class

of equally valuable composite signs it in fact is an abstraction. It does, however, not ignore characteristics of the reality. Rather it ignores irrelevant differences between its instances.

It is thus not suitable to regard a model as a mere abstraction. One should rather regard it as an abstraction injected into some “material” that, by its fundamental properties and particular instantiation, enables to make the intended use of the information as inherent in that abstraction and as required by the model’s purpose. Models are chiefly representations and successful models are handy representations.

It is not difficult to see more clearly some of the differences between conceptual modeling and abstraction. Let us understand an abstraction *A* of a thing *B* in an educated, but perhaps pre-scientific sense, as something that results from *B* by ignoring certain features of it. The claim that *A* is an abstraction of *B* can then be checked and either be confirmed or refuted. However, the claim that *A* is a model of *B* cannot be refuted. One might, at best, be able to show that *A* is not a good model of *B*, i.e., does not meet its purpose. Since no structural or other kind of similarity needs to exist between model and original one cannot falsify the mentioned utterance. Also, a representation may be a model instance for one community but not for another. Even if a representation is a model instance for each of two different communities, these communities may associate to it different originals.

Further light can be shed on the relationship between abstraction and representation by considering different modeling languages. Consider, for example, ER modeling vs. UML modeling. It is not difficult to see that, for example, the originals of UML use-case diagrams, class diagrams and state charts can be modeled with the ERM. The advantage of the UML is not its superior expressive power. Its advantage rather is the more intuitive approach to modeling, that rests on the peculiarities of the representations it instructs to use. To the UML, if compared to the ERM, “syntactical sugar” has been added that has nothing to do with the model original. It only has to do with the modelers’ abilities and desires to work on and with representations of a specific kind. In particular the visual formalism [7] used for creating and representing state charts makes the enriched model more appealing for human modelers than a related instance of the entity-relationship model.

Modeling for computerization often requires an inter-subjective meaning to be associated to the models. The models thus must be represented for that meaning to be sharable. They must be represented such, that making sense out of them is easy for the intended model users. The models thus must have characteristics that depend on the model users. Consequently, in general a model cannot be regarded as a mere abstraction of its original. Quite to the contrary, a model is a representation that is tailored towards its users. It is in fact one of the key difficulties in modeling to know how to best aid those users and to find the appropriate representations.

At the first glance it might not be that obvious what the material could be into which conceptual models are injected. However, conceptual models, in general,

must be implementable on a computer. Thus they need to have logical characteristics that allow for that. These certainly, in general, are not characteristics of the original. In fact it is quite possible to implement models of non-computable phenomena on a computer. The material in this case are languages, formalisms, hard and orgware, that permit the intended way of processing the model.

The database community strongly depends on the idea of models being abstractions injected into a suitable material. The well-known ANSI/SPARC-architecture [17] of database management systems, for example, introduces logical and physical data independence. These independencies mean that, as far as a related database application goes, the information content of the model is independent of how that model actually is implemented on a computer. Therefore, of all the applicable representations, those can be chosen, that are best at enabling the desired ways of handling the related information. The more appropriate representation may be the one that leads to queries being answered faster or the data quality being higher or similar.

4 Kinds of Normative References as Aided by Models

From Muller et al., [16], it can be derived that it is not uncommon for authors on conceptual modeling to consider models as descriptive or prescriptive. That way of conceptualizing models ascribes to them the capacity to describe or prescribe something else. I do not agree with this view because conceptual models are not auto-active and thus by themselves are incapable of describing or prescribing something. Moreover, that view, by implication, denies modeling communities the capacity to use the relationship between model and original thereof, that best fits the models's purpose.

Wieringa introduced the concept of “direction of fit” [24]. It addresses the kind of change that is needed in case an intolerable difference between a model and its original is detected. For “descriptive models” the direction of fit is from the model to the original. That means, if model and original do not fit each other well enough, then the model must be changed, so a fit of required quality will be achieved. With regard to “prescriptive models” the item that needs to be updated is the opposite one. The identification of the direction of fit is valuable. It does not address a model characteristic, however. Rather, it addresses the use that is made of the model. A model may be used to describe or prescribe something. I can easily exemplify the point by considering a typical descriptive model, such as a photograph of some important thing, such as an ancient temple. In case that temple would be destroyed one might want to rebuild it. Then one might find it helpful to take that photograph as a prescription. The photograph, that prior has served as a description of the temple, then starts to serve as the prescription of what to restore ore rebuild. To be descriptive or prescriptive is not an inherent characteristic of a model. It is instead a convention inherent in the reference established from that model to its original. A simple extension of the example shows that a model can be prescriptive with regard to one original and descriptive with regard to another. For example, the photograph of that temple

might be used to make a copy of it. In fact, design processes often require this kind of role-change. In software development a prototype can be given the status of a product specification. Prior to that the prototype might just have been a description of what could be implemented.

To focus on the different kinds of reference that are exploited in software development the term reference mode [11] has been coined. In that paper a number of such modes has been identified. The idea of the direction of fit extends quite largely into determining the various reference modes. For example, the idealizing reference mode is how software process models are related as a pre-image to the activity of software developers. Under ideal circumstances one would proceed exactly as specified by that model, however, the world is not ideal and therefore there can be good reason for deviating from the ideal way of doing things. Thus, even if there is a vast difference between what the model instructs to do and what was actually done, no corrective action might follow. In fact the deviation might not even be a problem. And, if there is such a problem then it might have to be resolved on an entirely different level. Another example is the constitutive reference mode. In this mode a model of an essentially unexplored part of the reality of a group of people is taken to replace in their reality that pre-discourse version of the thing. This is the mode typical for philosophical ontologies.

5 Resume

In this paper I have briefly presented and discussed a semantic theory of shared references. I have then used this theory to discuss two problems of conceptual modeling. The first of these was the relationship between abstracting and modeling. The second problem was the quality of the relationship between a model and its original. In writings about conceptual modeling, views on these problems have been put forward. I have shown that, contrary to some of those views, representation is key to modeling and in particular takes precedence over abstraction. Furthermore I have shown that, to be descriptive, prescriptive or similar, is not an inherent characteristic of a model but, instead, is a conventional characteristic of references carried out by modelers.

Acknowledgment. I thank Alexander Krumpholz and Bernhard Thalheim for their comments on an earlier version of this paper.

References

1. Booch, G., Rumbaugh, J., Jacobson, I.: *The Unified Modeling Language user guide*. Addison Wesley (1998)
2. Carnap, R.: *Der logische Aufbau der Welt*. Felix Meiner Verlag (1928)
3. Chen, P.: The Entity-Relationship model – toward a unified view of data. ACM ToDS 1(1) (1976)
4. Faroult, S., Robson, P.: *The art of SQL*. O'Reilly (2008)

5. Fleck, L.: *Entstehung und Entwicklung einer wissenschaftlichen Tatsache. Einführung in die Lehre vom Denkstil und Denkkollektiv.* Schwabe und Co., Verlagsbuchhandlung, Basel (1935)
6. Frigg, R., Hartmann, S.: Models in science. In: Zalta, E. (ed.) *The Stanford Encyclopedia of Philosophy* (fall 2012).<http://plato.stanford.edu>
7. Harel, D.: On visual formalisms. *Communications of the ACM* 31(5) (1988)
8. Harel, D., Rumpe, B.: Meaningful modeling: what's the semantics of "semantics"? *IEEE Computer* 37(10) (2004)
9. Kamlah, W., Lorenzen, P.: *Logische Propädeutik: eine Vorschule des vernünftigen Redens*, 2nd edn. Bibliographisches Institut. (1973)
10. Kaschek, R.: A little theory of abstraction. In: *Proceedings of Modellierung 2004*, GI Edition. Lecture Notes in Informatics, p. 45. Bonn (2004)
11. Kaschek, R.: Modeling ontology use for information systems. In: Althoff, K.-D., Dengel, A.R., Bergmann, R., Nick, M., Roth-Berghofer, T.R. (eds.) *WM 2005. LNCS (LNAI)*, vol. 3782, pp. 609–622. Springer, Heidelberg (2005)
12. Kralemann, B., Lattmann, C.: Models as icons: modeling models in the semiotic framework of Peirce's theory of signs. *Synthese* (September 2012), doi:10.1007/s11229-012-0176-x
13. Lockemann, P., Mayr, H.: *Rechnergestützte Informations systeme*. Springer (1978)
14. Marx, K.: Theses on Feuerbach, vol. 1845 (2012),
<http://www.marxists.org/archive/marx/newlineworks/1845/theses/theses.htm> (accessed on December 3)
15. Morris, C.: *Signs, language and behavior*. Prentice-Hall, Inc. (1946)
16. Muller, P.-A., Fondement, F., Baudry, B., Combemale, B.: Modeling modeling. *Softw. Syst. Model* 11 (2012)
17. Ramakrishnan, R., Gehrke, J.: *Database management systems*, 3rd edn. McGraw-Hil (2003)
18. Seidewitz, E.: What models mean. *IEEE Software* 20(5) (2003)
19. Stachowiak, H.: *Allgemeine Modelltheorie*. Springer (1973)
20. Teorey, T., Lightstone, S., Nadeau., T., Jagadish, H.V.: *Database modeling and design: logical design*, 5th edn. Morgan Kaufmann (2011)
21. Thalheim, B.: Syntax, semantics and pragmatics of conceptual modeling. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) *NLDB 2012. LNCS*, vol. 7337, pp. 1–10. Springer, Heidelberg (2012)
22. Thalheim, B.: The definition of the (conceptual) model. In: *Proceedings of EJC 2013. IOS Press, Amsterdam* (2013)
23. Wand, Y., Weber, R.: Research commentary: information systems and conceptual modeling – a research agenda. *Information Systems Research* 13(4) (2002)
24. Wieringa, R.: Algebraic foundations for dynamic conceptual models. PhD thesis, Free University of Amsterdam, Amsterdam (May 1990)

Are Conceptual Models Concept Models?

Chris Partridge¹, Cesar Gonzalez-Perez², and Brian Henderson-Sellers³

¹ Brunel University, Uxbridge, Middlesex UB8 3PH, United Kingdom

² Institute of Heritage Sciences (Incipit), Spanish National Research Council (CSIC), Santiago de Compostela, Spain

³ Faculty of Engineering and Information Technology, University of Technology, Sydney, P.O. Box 123, Broadway, NSW 2007, Australia
chris.partridge@brunel.ac.uk,
cesar.gonzalez-perez@incipit.csic.es,
brian.henderson-sellers@uts.edu.au

Abstract. The conceptual modelling community not only has no clear, general agreement on what its models model, it also has no clear picture of what the available options and their implications are. One common claim is that models represent concepts, but there is no clear articulation of what the concepts are. This creates theoretical problems; for example, it is difficult to justify the accuracy of meta-models. It also creates practical problems; practitioners building a model of the ‘concept’ of a business will rationalise their decisions differently from those modelling the business itself, making resolving disagreement difficult. In contrast, philosophy has been researching this area for millennia and has developed, at the high level, a clear picture of the semantic landscape, particularly for concepts. This presents an opportunity to provide the conceptual modelling community with a ready-made framework for its semantic options. We start exploiting this opportunity, developing here an initial framework.

Keywords: concepts, modelling, philosophy, semantics, ontology, representation, meaning, type nominalism, abstract object nominalism.

1 Introduction

For disciplines and practices to evolve, their foundations also need to evolve. The conceptual modelling community’s current practices for engineering information systems are built upon heterogeneous, unarticulated (or, at least, not clearly articulated) assumptions about the structure and nature of the components of the semantic relation that binds the model to its domain. Providing a more thought-through picture of what possible options there are for these components will, in the short term, make it clearer what commitments are currently being made and their implications. This, in the longer term, should lead to clearer and stronger foundations for future development.

The semantic components are an area of research in a number of disciplines. However, there is one discipline, philosophy, that has focused on looking at ‘what’ these components could be (in contrast to, for example, how they might behave). What

marks out this philosophical research is an attempt to chart the conceptual landscape; without, initially at least, a commitment to a particular view. This provides a suitable framework for understanding how conceptual modelling's semantic components could, and should, be grounded.

Though undoubtedly members of the conceptual modelling community have been influenced by philosophical ideas – either directly or through cultural osmosis – philosophy has actually had little real overt influence upon conceptual modelling development. In the last decade, the conceptual modelling community has shown some interest in, and exploitation of, philosophical work in ontology to help in understanding the model's domain [1-3] but its research into the semantic relation between the model and the domain has been mostly a home-grown, philosophy-free (or philosophy-light) activity. Consequently, this opportunity for inter-disciplinary fertilisation is un-exploited.

In this paper, we intend to start this exploitation. We outline a preliminary framework for organising the current range of views on semantics in the conceptual modelling community based upon research in philosophy. We look at philosophy's analysis of the potential motivations for its classifications as well as its analysis of the issues they face. This starts to make sense of the variety of views in conceptual modelling, albeit from a philosophical viewpoint. It provides a broad-brush structure that can be used as a roadmap for some of the basic choices that can be made for the semantic foundations of conceptual modelling; and a framework within which more fine-grained positions can be articulated.

2 Background

Conceptual modelling has not yet developed a clear picture of its semantics, having emerged relatively recently from physical data modelling in response to a requirement to ‘abstract’ away from particular physical implementations [4]. The physical data model's semantics bind it to the target system. Conceptual models have a different semantics, one that binds them to the domain being represented by information in the target system; separating the two semantics can be a challenge [5].

Philosophy is an ancient discipline that has researched semantics from its inception resulting in a complex picture. Within philosophy, a premium is put on exploring the full range of choices. There is a tradition of developing sophisticated structures to finesse the various issues encountered. This has led to a wide variety of often very sophisticated solutions. This is sensible in a pure scientific discipline, but is inappropriate in a pragmatic engineering discipline such as conceptual modelling. The objective of this paper is to extract from the sophisticated philosophical research a pragmatic framework suitable for the engineering goals of conceptual modelling.

There are three scope setting points that it is useful to clarify at the start. Firstly there is a need to clarify the sense of semantics used in this paper. The philosopher David Lewis described the distinction thus: “I distinguish two topics: first, the description of ... abstract semantic systems whereby symbols are associated with aspects of the world; and, second, the description of the psychological and sociological

facts whereby a particular one of these abstract semantic systems is the one used by a person or population. Only confusion comes of mixing these two topics.” [6] (p. 19). To avoid the confusion to which Lewis refers, we would like it to be clear that this paper concerns the first topic, associating symbols with the world. In conceptual modelling terms, this can be translated as asking what elements would appear in a model of the semantic relation. This is, of course, not to say research on the second topic would not be useful, merely that it is not in our scope.

Secondly, there is a need to recognize that conceptual modelling is a specialized language [7] (p. 105); one where the semantic relation is, to an extent, constructed and where there is a requirement that we construct a clear relation.

Thirdly, there are interesting questions raised about the nature of models, because they typically have several copies. From a philosophical perspective, this seems to be analogous to, if not the same as, the type-token distinction [8] (sec. 4.537). While researching this would be a useful exercise, it is not a topic for this paper.

3 Semantics of Conceptual Models

Prima facie, the name ‘Conceptual Modelling’ suggests that *concepts* are involved in some way; it could be that concepts are being modelled or that the models work in an analogous way to concepts. Conceptual modelling practice is no real help in answering this. For example, the conceptual modelling languages (e.g. ER, UML) can accommodate a variety of views, avoiding imposing any particular view, even within a single model. This paper explores how different views might fit into a foundation for conceptual modelling.

For the foundations of conceptual modelling, a core question is: What kinds of thing are being modelled in a conceptual model? Examination of the literature and interrogation of practitioners shows a broad acceptance that models are ‘about’ a domain. It also gives, very broadly speaking, two explanations of the structure of this ‘about’ relation:

1. Concept-mediated-semantics: The elements in a conceptual model represent concepts. These concepts then represent things in the domain. The concepts mediate between the model and the domain [9]. Note that the use of the term ‘concept’ in the mediating sense is commonplace. Good examples of this outside conceptual modelling are the foundational ISO standards for terminology [10] and [11], which are built around this term.
2. Direct-domain-semantics: The elements in a conceptual model directly represent things in a domain [12].

Yet others claim a combination of these two views: that the elements represent (in different ways) both concepts and things in a domain (‘the real world’) [3,13], where concepts play a mediating role. This dual semantics originates from [14] and is often described as the Ullmann triangle. Popular in a number of areas including linguistics and conceptual modelling, there are few if any references to it in philosophical writings. We shall not analyse this position separately but regard it as an amalgamation of the two options above.

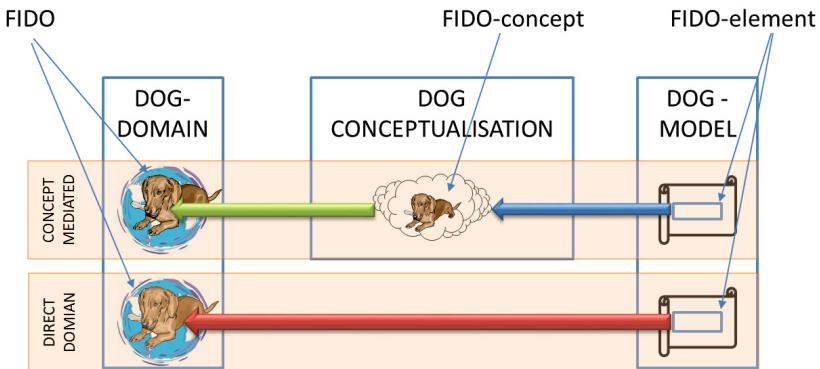


Fig. 1. Simple schematic of the two semantic structures

The (very) simple model in Fig. 1 illustrates the core differences between the two options. In one, the domain is referred to directly; in the other, concepts mediate. The direct-domain option is clearly structurally simpler. Consequently, the benefits of a concept-mediated semantics needs to outweigh the loss of simplicity.

In philosophy, concept-mediated-semantics have a long tradition stretching back to Aristotle [15] (1.16a4), but most clearly enunciated in Locke [16] (III, II), who clearly stated that only mental concepts can represent things whilst words represent mental concepts. This was a key part of Locke's theory of concepts as ideas, which were an essential part of the mechanics of representation. However, this tradition has largely died out in modern philosophy, and there has been a shift towards the acceptance of direct-domain-semantics, particularly in the philosophy of language and logic. This acceptance of direct-semantics is especially true in philosophy of logic when discussing formal languages – and, as noted earlier, conceptual modelling can be regarded as a formal language. In these, it is usual to adopt a simple direct semantic relation, known as the ‘intended interpretation’.

4 Mediating Concepts

The nature of concepts is a grey area in conceptual modelling, one where it would be useful to shed some light. Within the current philosophy of mind there is substantial research into what a concept is – understanding this is key to understanding concept-mediated-semantics.

To foster this understanding, this section firstly distinguishes three ‘mediating concept’ positions (divided into two categories) based upon their view of what a concept is. It then distinguishes two positions based on what concepts represent. These positions are:

1. Concepts-as-mental
 - a. Concepts-as-representations; Concepts are mental representations; normally within a wider representational theory of the mind, where thinking is regarded as a system of representation. The concept DOG is a general representation of a dog.

- b. Concepts-as-abilities; Concepts are abilities. The concept DOG is an ability to tell dogs from non-dogs [17].
- 2. Concepts-as-mind-independent
 - a. Concepts-as-meanings; Concepts are meanings, or more technically, they are Fregean senses [18]. The concept DOG is the meaning of the general term ‘dog’.

The two categories separate the location of concepts into mind-dependent (mental) and mind-independent camps. Within the mental camp, the two positions take different positions on whether concepts represent.

Finding an uncontentious position on concepts is difficult. The concepts-as-mental category faces a simple explanatory issue. If concepts are mental, then each concept must belong to only one mind and the concept is private to that mind; yet there is typically more than one person involved with a conceptual model during its life. Furthermore, as models grow, they become so big that no one person can fit the whole conceptualisation into their mind. In practical terms, this does not sit well with the proposed mediating role. One may loosely speak of people sharing the same concept; but strictly speaking, this is impossible. One may relax the requirement to sufficient similarity but this brings its own problems [19].

Concepts-as-representations is the current mainstream view in philosophy [20] and cognitive science [21] as well as being common in conceptual modelling. It also has a long history: Locke [16] and Hume [22] were early advocates. However, if concepts are the ‘words’ of a mental language, then they would seem to be on a par with conceptual modelling languages. It would seem they are ‘yet another language’. To make sense, mental languages need to have special properties missing in conceptual modelling languages.

The concepts-as-abilities position dispenses with mental representations. This can be motivated by a deep scepticism about whether mental representations (that is, concepts-as-representations) can do the work they need to – a view that traces back to Wittgenstein [23]. The core argument is that mental representations *as representations* reintroduce the very sorts of problems they are supposed to explain. If the way we process an external representation is to create a corresponding mental representation, then we presumably have to create a further mental representation of this in order to process it, thus leading to an endless regress. From a conceptual modelling perspective, if one takes this position then probably one has to adopt a direct-domain-semantics, since concepts-as-abilities do not have quite the right features for mediation.

There are other ramifications. It is common practice to involve domain experts in the building of conceptual models. If one adopts a concepts-as-representations position, then one would expect the building of the model to be a transcribing of the mental representations into the model – a translation from one representation to another. However, if the concepts-as-abilities position is in fact true, then the translation from expert ability to representation will be less straightforward [24], thus having a significant impact on how to organise and budget the modelling.

The concepts-as-meanings position is a view about semantics rather than the mind. It is often closely linked with Frege's [18] view on sense and hence concepts-as-meanings are sometimes called senses – though it can be terminologically confusing since Frege did not use the term 'concept' for senses but for something else.

Concepts-as-meanings mediate between representations (conceptual models) and the represented domain. Peacocke [25] (p. 169) argues that as there could be concepts-as-meanings that are never thought – or too complex to ever be thought – they cannot be mental. So the mental plays no explanatory role.

A key feature of the concepts-as-meanings criterion of identity is its fine-grainedness; its ability to make very fine distinctions. Using an example from Peacocke [26]; the concepts 'Samuel Clemens' and 'Mark Twain' are different even though they refer to the same person. One might wonder whether these different concepts-as-meanings are more fine-grained than reality; that one will need to systematically create two objects in a model referring to one thing in the world. If the goal is to directly represent the domain, then concepts-as-meanings appears to be the wrong tool.

There is also a need to explain how we know about concepts-as-meanings; what the connection is between them and the mental. It is claimed that they can be grasped by human minds; that the same concept can be 'grasped' by different minds, though there is little detail of how this works. This avoids some of the issues with sharing concepts that concepts-as-representations face as noted above.

Foundation models also need to represent the domain, but there are ontological ("what") questions about the domain that need to be answered, which we mention briefly. A simple semantic assumption is that the elements in the model map one-to-one onto the things in the domain – either directly or via concepts. However, as models often contain general elements, such as DOG, this implies that the world contains corresponding types. However, if one is a nominalist about types, then one thinks that the general elements in the model (or general concepts in the conceptualisation) map onto many things in the domain; Locke [16] (III.iii.11) was an early nominalist. The type-realist assumes that they map one-to-one onto types. While there are a variety of motivations for the nominalist position, one clear advantage in a concept-mediated-semantics is parsimony (Ockham's razor); it eliminates the need for types. Where the realist has an ontic 'instance' relation between types and particulars, the nominalist has a semantic relation between concepts (or terms) and particulars. Within philosophy, there are adherents to both the nominalist and realist positions, as well as a number of varieties of each. Within conceptual modelling, there is little or no research to decide whether one of these is preferable for its adoption.

In modern philosophy, there is a widely adopted view that every entity is either concrete or abstract - a fundamental distinction. There is broad agreement on clear cases: the number one is abstract and Fido the dog is concrete. A common metric is that abstract objects are not spatially or temporally situated and, hence, are causally inert.

Abstract object nominalism takes the view that abstract objects cannot exist whereas abstract object realism takes the view that they can [27]. As concepts-as-meanings are abstract, anyone who adopts them needs to be some form of abstract object realist. However, this raises issues. For example, it appears difficult to explain

how anyone can know about abstract objects if they are not situated anywhere and cannot cause anything [28]; especially as they cannot be directly studied. This problem is familiar to practitioners of conceptual modelling, where securing agreement on a model of abstract objects can be extremely difficult – and often descends into comparisons of competing intuitions.

5 An Emerging Framework: Discussion and Conclusions

A preliminary framework has emerged from the analysis. Two major options have been identified for the semantic structure: concept-mediated or direct-domain semantics. Within the former, there are further options of concept ontology: concepts as representations, abilities or meanings. There are two associated ontological choices between different types of realism and nominalism: one for types, the other abstract objects. These choices, apart from abstract object realism-nominalism, have visible effects on the structure of the semantic relation – as illustrated for a sample of general elements in Table 1.

Table 1. Semantic Structure: Model relations for general elements

Choice	Source	Target	Relation Type	Cardinality
Directed -- Realism	Model Element	Type	semantic	1:1
	Type	Particular	ontic	1:M
Directed -- Nominalism	Model Element	Particular	semantic	1:M
Mediated - Meaning - Realism	Model Element	Concept	semantic	1:1
	Concept	Type	semantic	M:1*
	Type	Particular	ontic	1:M
Mediated - Meaning - Nominalism	Model Element	Concept	semantic	1:1
	Concept	Particular	semantic	1:M
Mediated - Representation - Nominalism	Model Element	Concept	semantic	1:M
	Concept	Type	semantic	M:1
	Type	Particular	ontic	1:M

* - it has been assumed, as seems to be the case, that concepts-as-meanings are finer grained than reality.

It may help to ground this structure with an example from UML; where a Class DOG may represent: (i) a single type or (ii) a single concept-as-meaning; or (iii) a number of concepts-as-representation (one for each mind).

This framework will need further development, maybe even substantial revision. However, as it now stands, it can be used to clarify existing positions. As it develops, it can be used as a basis for narrowing down and refining the choices of the community eventually to a particular set of choices that makes the most sense for conceptual modelling.

The various issues raised earlier for the concept-mediated positions suggest that there are some hurdles to overcome before arriving at a clear picture; and, given that philosophical research has not provided this, it may be that they turn out to be unsuitable candidates for the foundations. Direct-domain semantics may be a more suitable candidate. This position has some champions; for example, Smith [2] argues strongly against concept-mediated semantics.

6 Further Research Work

As we have noted a couple of times, this paper is intended to be the start of a process of exploitation of relevant research in philosophy. Our analysis identified a number of areas that were not in the scope of this paper, but looked as if they could usefully be explored, of which the following four stand out.

In conceptual modelling, much of the loose talk of concepts seems to have no substance; it does not translate into practice. One way of understanding this is taking a fictionalist attitude to such talk. Fictionalism about discourse is the view that it is not to be taken literally but as a sort of useful metaphor [29]. If this is true, then criticising it for not being literally true is to misunderstand the situation.

There is work in the philosophy of mind that aims to extend the boundaries of the mental, first to the boundaries of the body (the embodied mind) and then beyond (the extended mind) [30]. This seems particularly relevant to conceptual modelling, opening up the possibility of the model being subsumed into the mental representation wherein the model elements are the concepts involved.

Much conceptual modelling takes place in teams. There is a growing interest in social epistemology [31]; in understanding how groups of people can know. This may give conceptual modelling an insight into how a team knows. Combined with the notion of the extended minds, this could explain how the model could be the embodiment of social concepts.

There is much talk of abstraction in conceptual modelling as a vital mental process in the building of models e.g. [32]. The term is used in the sense of omitting detail, though it is not always clear whether this is a mental or formal process. Historically, in philosophy it played an important role in British Empiricism. It played a central role in Locke's philosophy of psychology where the mind is described as capable of creating new mental images by omitting detail from existing ones. This view has been a subject of devastating criticism in contemporary philosophy, Frege being an early example – see [17] (p. 85). Given the prevalence of something akin to the simple Lockean notion in conceptual modelling, it would seem to make sense to expose these criticisms and revisions to the community.

Acknowledgements. We wish to thank Sergio de Cesare, Mesbah Khan and Andy Mitchell for their detailed reviews.

References

1. Partridge, C.: Business Objects: Re - Engineering for Re - Use. Butterworth Heinemann, Oxford (1996)
2. Smith, B.: Beyond Concepts: Ontology as Reality Representation. In: Proceedings of the Third International Conference (FOIS 2004), pp. 73–84. IOS Press, Amsterdam (2004)
3. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models, CTIT PhD Thesis Series, no. 05-74 (2005)
4. Mylopoulos, J.: Conceptual Modelling and Telos. In: Loucopoulos, P., Zicari, R. (eds.) Conceptual Modeling, Databases, and Case: An Integrated View of Information Systems Development, pp. 49–68. John Wiley & Sons, New York (1992)
5. Lycett, M., Partridge, C.: The Challenge of Epistemic Divergence in IS Development. *Commun. ACM* 52, 127–131 (2009)
6. Lewis, D.: General Semantics. *Synthese* 22, 18–67 (1970)
7. Frege, G.: Conceptual Notation, and Related Articles. Oxford Scholarly Classics. Clarendon Press (1972)
8. Peirce, C.S.: Collected Papers of Charles Sanders Peirce. Harvard University Press, Cambridge (1932)
9. Wieringa, R.: Real-World Semantics of Conceptual Models. In: Kaschek, R., Delcambre, L. (eds.) The Evolution of Conceptual Modeling. LNCS, vol. 6520, pp. 1–20. Springer, Heidelberg (2011)
10. ISO: ISO 1087-1:2000 - Terminology Work – Vocabulary – Part 1: Theory and Application 41 (2000)
11. ISO: ISO 704:2009 Terminology work–Principles and Methods. International Organization for Standardization 65 (2000)
12. Wand, Y., Storey, V.C., Weber, R.: An Ontological Analysis of the Relationship Construct in Conceptual Modeling. *TODS* 24, 494–528 (1999)
13. Henderson-Sellers, B., Eriksson, O., Gonzalez-Perez, C., Ågerfalk, P.J.: Ptolemaic Metamodelling? The Need for a Paradigm Shift. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., Bettin, J. (eds.) Research Directions in Domain Engineering. Springer, Berlin (in press, 2013)
14. Ogden, C.K., Richards, I.A.: The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism, Harcourt, Brace, New York (1923)
15. Aristotle: The Categories; on Interpretation. Harvard University Press; W. Heinemann, Cambridge, Mass. London (1983)
16. Locke, J.: An Essay Concerning Human Understanding. Clarendon Press, Oxford (1975) (First published 1690)
17. Dummett, M.: The Seas of Language. Clarendon press, Oxford (1993)
18. Frege, G.: Über Sinn Und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik* 100, 25–50 (1892)
19. Davis, W.A.: Meaning, Expression and Thought. Cambridge Studies in Philosophy. Cambridge University Press, Cambridge (2003)
20. Carruthers, P.: Phenomenal Consciousness: A Naturalistic Theory. Cambridge University Press, Cambridge (2000)
21. Pinker, S.: The Language Instinct: The New Science of Language and Mind. Penguin, London (1995)
22. Hume, D.: A Treatise of Human Nature. John Noon, London (1739)
23. Wittgenstein, L.: Philosophical Investigations. Blackwell Publishing, Oxford (1953)

24. Partridge, C., Lambert, M., Loneragan, M., Mitchell, A., Garbacz, P.: A Novel Ontological Approach to Semantic Interoperability between Legacy Air Defence Command and Control Systems. *International Journal of Intelligent Defence Support Systems* 4, 232–262 (2011)
25. Peacocke, C.: Rationale and Maxims in the Study of Concepts. *Noûs* 39, 167–178 (2005)
26. Peacocke, C.: *A Study of Concepts. Representation and Mind*. The MIT Press, Cambridge (1992)
27. Field, H.H.: *Science without Numbers: A Defence of Nominalism*. Princeton University Press, Princeton (1980)
28. Benacerraf, P.: Mathematical Truth. *The Journal of Philosophy* 70(19), 661–679 (1973)
29. Yablo, S.: Go Figure: A Path through Fictionalism. *Midwest Studies in Philosophy* 25, 72–102 (2001)
30. Clark, A.: *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*. Oxford University Press, New York (2008)
31. Goldman, A.I.: Foundations of Social Epistemics. *Synthese* 73, 109–144 (1987)
32. Kaschek, R.: A Little Theory of Abstraction. In: *Proceedings of Modellierung 2004*, pp. 75–92. Gesellschaft für Informatik, Bonn (2004)

Visual Modeling of Business Process Compliance Rules with the Support of Multiple Perspectives*

David Knuplesch¹, Manfred Reichert¹, Linh Thao Ly¹,
Akhil Kumar², and Stefanie Rinderle-Ma³

¹ Institute of Databases and Information Systems, Ulm University, Germany

{david.knuplesch,manfred.reichert,thao.ly}@uni-ulm.de

² Smeal College of Business, Pennsylvania State University, PA, USA

akhilkumar@psu.edu

³ Faculty of Computer Science, University of Vienna, Austria

stefanie.rinderle-ma@univie.ac.at

Abstract. A fundamental challenge for any process-aware information system is to ensure compliance of modeled and executed business processes with imposed compliance rules stemming from guidelines, standards and laws. Such compliance rules usually refer to multiple process perspectives including control flow, time, resources, data, and interactions with business partners. On one hand, compliance rules should be comprehensible for domain experts who must define and apply them. On the other, they should have a precise semantics such that they can be automatically processed. In this context, providing a visual compliance rule language seems promising as it allows hiding formal details and offers an intuitive way of modeling. So far, visual compliance rule languages have focused on the control flow perspective, but lack adequate support for the other perspectives. To remedy this drawback, this paper provides an approach that extends visual compliance rule languages with the ability to consider data, time, resources, and partner interactions when modeling business process compliance rules. Overall, this extension will foster business process compliance support in practice.

Keywords: business process compliance, compliance rule graphs, business process modeling, business intelligence.

1 Introduction

During the last decade, numerous approaches for ensuring the correctness of business processes have been discussed [1, 2]. Most of them focus on syntactical correctness and process model soundness (e.g., absence of deadlocks and lifelocks). However, business processes must also comply with semantic rules stemming from domain-specific requirements such as corporate standards or legal regulations [3]. Summarized under the notion of *business process compliance*, existing

* This work was done within the research project C³Pro funded by the German Research Foundation (DFG), Project number: RE 1402/2-1, and the Austrian Science Fund (FWF) under project number: I743.

approaches have mostly considered compliance issues related to the control flow perspective of single processes. By contrast, cross-organizational scenarios characterized by interacting and collaborating business processes of various parties have not been properly considered so far [4]. Furthermore, compliance requirements for both local and global process scenarios do not only concern control flow and interactions between business partners (i.e. messages exchanged), but also refer to time, resources, and data [5–8]. As examples, consider the compliance rules in Tab. 1, which are imposed on a cross-organizational process scenario involving the two business partners *reseller* and *manufacturer*. In particular, as shown by the highlighted terms in Tab. 1, the rules that arise in practice should be able to describe aspects of interaction, time, resource and data as they relate to a business process. Hence, these various perspectives of a business process should be modeled to support compliance.

Compliance rule c_1 considers a pair of interactions between a reseller and manufacturer (*request* and *reply*) after a particular point in time (*3rd January, 2013*) as well as the maximum time delay between them (*within three days*). The data perspective of compliance rules is emphasized by compliance rule c_2 of the manufacturer. It forbids changing an *order* after having started the corresponding *production* task. Compliance rule c_3 in turn, combines the interaction, time, and data perspectives. Finally, compliance rule c_4 introduces the resource perspective (*member of the order processing department* and *another member of the same department with supervisor status*). In addition, c_4 considers the data perspective (e.g. *new customer* and *total amount greater than €5,000*) and the time perspective (*at most three days*). Particularly c_4 shows that the different perspectives might be relevant for the same rule and hence cannot be considered in an isolated manner.

Comparing c_4 and c_2 with c_1 and c_3 , one can further notice two different viewpoints: c_4 and c_2 are expressed from the viewpoint of the manufacturer (i.e., local view), while c_1 and c_3 reflect a global view. Note that such distinction between local and global views is common to cross-organizational collaboration scenarios

Table 1. Examples of compliance rules for order-to-delivery processes

c_1	Any <i>request</i> sent from the reseller to the manufacturer <i>after January 3rd, 2013</i> should be <i>replied</i> by the manufacturer <i>within three days</i> .
c_2	After starting the production related to a particular <i>order</i> , the latter must not be changed anymore.
c_3	When the manufacturer <i>sends a bill</i> with an <i>amount lower than €5,000</i> to the reseller, the latter must make the payment <i>within 7 days</i> .
c_4	After receiving a production request message from the reseller, which refers to a <i>new customer</i> and has a <i>total amount greater than €5,000</i> , the solvency of this customer must be checked by a <i>member of the order processing department</i> . Based on the result of this check, <i>another member of the same department with supervisor status</i> must approve the request. Finally, the approval result must be sent to the reseller <i>at most three days</i> after receiving the original request.

not only in the context of process compliance. For example, BPMN 2.0 provides collaboration and choreography diagrams to express these different viewpoints.

Several approaches for formally capturing compliance requirements at different abstraction levels (e.g., temporal logics [9]) exist to enable the automatic verification of compliance of business processes with such rules. As the use of formal languages for compliance rule specifying might become too intricate, rule patterns [6, 8], which hide formal detail from rule modelers, have been proposed. Furthermore, a few approaches also consider more advanced issues like, e.g., the use of data conditions in the context of compliance requirements. However, existing approaches are usually restricted to a specific subset of rule patterns. In this context, rule languages, employing visual notations like the compliance rule graph approach [10] or BPSL [11], provide an alternative as they combine an intuitive notation with the advantages of a formal language. However, our meta-analyses and case studies, we conducted in domains like higher education, medicine and automotive engineering, have revealed that these visual compliance rule languages still lack support for the time, data, and resource perspective of business processes. Our analyses have further shown that existing compliance rule notations do not consider cross-organizational scenarios with interacting partners [4]. Overall, in our meta-analyses and case studies, we elicited the following fundamental requirements for visual compliance rule languages:

- In addition to the control flow perspective, the data, resource and time perspectives of compliance requirements must be properly captured.
- To not only consider process orchestrations, but cross-organizational scenarios as well, it becomes necessary to express the interaction perspective with compliance rule languages.
- To provide tool support for both the modeling and verification of compliance rules, their syntax as well as semantics must be formalized.

To cope with the shortcomings discussed above, we introduce extensions for visual compliance rule modeling supporting the data, time, and resource perspectives of business processes. More precisely these extensions are proposed for the compliance rule graph (CRG) language we developed in earlier work [10, 12]. However, the major concepts we propose may be applied to other compliance rule languages as well. Another fundamental contribution is the ability of our *extended compliance rule graph language* to specify compliance requirements for cross-organizational scenarios (i.e. processes choreographies) as well. For this purpose, we additionally introduce concepts that allow defining compliance rules in respect to message flows and partner interactions. Altogether, the visual compliance rule language developed in this paper allows capturing compliance requirements at an abstract level, while at the same time it enables the specification of verifiable compliance rules in the context of cross-organizational scenarios.

The remainder of this paper is structured as follows: Sect. 2 discusses related work. In Sect. 3, we introduce the data, time, resource, and interaction perspective of compliance rules. Our extensions of the CRG language regarding the support of these perspectives, the *extended compliance rule graphs* (eCRG), are described in Sect. 4. To validate our approach, we present a proof-of-concept

prototype and outline the results of a pattern-based evaluation in Sect. 5. Sect. 6 concludes the paper and provides an outlook on future research.

2 Related Work

Recently modeling issues related to the interaction, time, resource, and data perspectives of business processes have been addressed in addition to the control flow perspective (e.g., [13–19]).

The integration of business process compliance throughout the entire process lifecycle has been discussed in [12, 20–22]; [23] examined compliance issues in the context of cross-organizational processes developing a logic-based formalism for describing both the semantics of normative specifications and compliance checking procedures. This approach allows modeling business obligations and regulating the execution of business processes. In turn, [24] introduced a semantic layer that interprets process instances according to an independently designed set of internal controls. Furthermore, there exist approaches using semantic annotations to ensure compliance [25]. An approach checking the compliance of process models against semantic constraints as well as ensuring the validity of process change operations based on Mixed-Integer Programming formulation is proposed in [26]. It introduces the notions of *degree of compliance*, *validity of change operations*, and *compliance by compensation*. [6] uses alignments to detect compliance violations in process logs. To verify whether compliance rules are fulfilled by process models at design time, many approaches apply model checking techniques [9, 11, 27–29]; some of them address the data and time perspectives as well. Further approaches for verifying compliance apply the notion of *semantic congruence* [30] or use *petri-nets* [31] and consider the data and time perspectives as well.

The approach described in [27, 32] for visually modeling compliance considers the control flow and data perspectives. It is based on linear temporal logic (LTL), which allows modeling the control flow perspective based on operators like *next*, *eventually*, *always*, and *until*. Finally, visual approaches for compliance rule modeling exist [11, 10, 33]. However, they focus on control flow and partly the data perspective, but ignore the other perspectives mentioned.

3 Compliance Perspectives

As noted above, compliance rules cannot be expressed completely by referring only to the control flow perspective of a business process. In [5–8], the importance of the time, resources, and data perspectives are emphasized. The need for ensuring compliance in the context cross-organizational scenarios is raised in [4]. Before introducing the visual notation of the eCRG language, we describe the compliance perspectives as well as related language concepts in more detail. The latter have been elicited through our analyses and case studies. Fig. 1 provides an overview of the perspectives we consider and characterizes their main features.

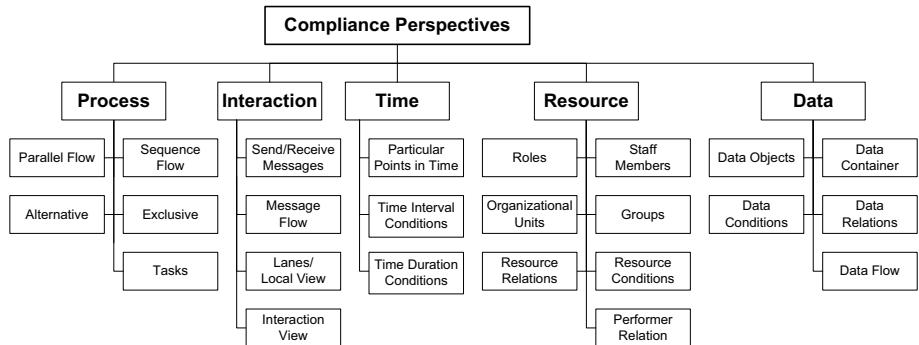


Fig. 1. Compliance perspectives

Process Perspective. The process (i.e. control flow) perspective of compliance rules is the most fundamental one. It comprises elements for expressing both the occurrence and presence (i.e., *exclusive*, *alternative*) of *tasks* as well as their ordering (i.e., *sequence flow*, *parallel flow*).

Interaction Perspective. In cross-organizational scenarios, compliance rules require particular elements for sending and receiving *messages*. *Message flows* correlate the sending and receiving of messages. Further, *lanes* express *local views* of the different partners on the different tasks to be performed. In turn, the *interaction view* focuses on the global sequence of interactions (i.e., messages exchanged). Compared to BPMN 2.0, *local views* correspond to *collaboration diagrams* and *interaction views* to *choreography diagrams*.

Time Perspective. Time support for compliance rules is tripartite: First, particular points in time may have to be expressed (e.g. *Monday, 3rd January 2013*). Second, conditions on the time intervals between events, tasks and points in time require support. Third, the duration of tasks may have to be constrained.

Resource Perspective. The resource perspective requires concepts for expressing constraints on *resources*. We select *staff member*, *group*, *organizational unit*, and *roles* as common concepts of organizational models. However, this list can be extended easily. The *performer relation* constrains the performer of a particular task. In turn, *resource conditions* and *relations* may be used to specify and constrain resources on a finegrained level.

Data Perspective. The data perspective comprises concepts for expressing data-aware compliance rules. Thereby, *data containers* refer to process data elements or global data stores. By contrast, *data objects* refer to particular data values and object instances. *Data flow* defines which process tasks read or write which data objects or data container. To constrain data container, data objects, and data flow, *data conditions* and *data relations* may be used.

In the following, required language extensions are presented taking the compliance rule graph (CRG) language as basis (cf. Sec. 4.1). However, these extensions may be applied to other compliance rule languages as well, since they are independent from particular properties of CRGs.

4 Extended Compliance Rule Graphs

This section introduces *extended Compliance Rule Graphs (eCRG)* - a visual notation for compliance rule modeling covering the process, interaction, time, resource, and data perspectives (cf. Section 3). Sect. 4.1 introduces fundamentals of CRGs, while its extensions are subsequently introduced step-by-step.

4.1 Fundamentals of Compliance Rule Graphs

The compliance rule graph (CRG) language [10, 12] allows visually modeling compliance rules whose semantics is defined over event traces. More precisely, a CRG is an acyclic graph consisting of an *antecedence pattern* as well as at least one related *consequence pattern*. Both patterns are modeled using *occurrence* and *absence* nodes, which indicate the occurrence or absence of events (e.g., related of the execution of a particular task). Edges between such nodes indicate control flow dependencies. As illustrated in Fig. 2, a trace is considered as compliant with a CRG iff for each match of the antecedence pattern there is at least one corresponding match of every consequence pattern. Further, a trace is considered as *trivially compliant* iff there is no match of the antecedence pattern. For example, the CRG from Fig. 2 expresses that for each *B* not preceded by an *A*, there must occur a *D*, which is not preceded by any *C* also preceding the respective *B*.

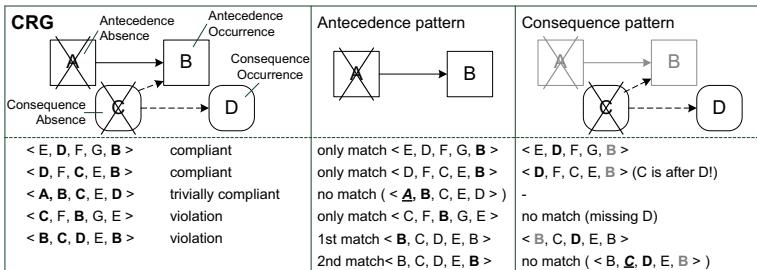


Fig. 2. CRG example and semantics over execution traces

In the following, we introduce the eCRG language, which is based on CRGs. Note that in addition to nodes and connectors (i.e., edges) as fundamental elements of graphs, eCRGs further support *attachments*. Attachments represent constraints to the nodes or edges they are attached to. Further, eCRGs may contain *instance nodes* representing particular instances, which exist independently from the respective rule (e.g. a particular employee *Mr. Smith*, date *3rd January 2013*, or role *supervisor*). Hence instance nodes are neither part of the antecedence nor the consequence pattern.

4.2 Process Perspective

The eCRG elements for modeling the process (i.e. control flow) perspective of compliance rules are introduced in Fig. 3. Since the extensions are based on

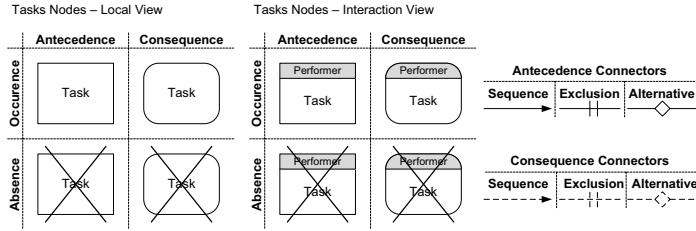


Fig. 3. eCRG elements of the process perspective

the CRG language, there are four different task elements, i.e., *antecedence occurrence*, *antecedence absence*, *consequence occurrence*, and *consequence absence*. These allow expressing whether or not particular tasks must be executed. In addition, two different kinds of *sequence flow connectors* are provided that may be used to constrain the execution sequence of tasks. Note that the absence of sequence flow indicates parallel flow. To clearly distinguish between *start-start*, *start-end*, *end-start*, and *end-end* constraints on the execution sequence of tasks, sequence flow edges are either connected to the right or left border of a task node. Furthermore, *exclusive connectors* denote mutual exclusion of tasks. *Alternative connectors* express that at least one of the connected tasks must occur. Note that exclusive as well as alternative connectors may only connect nodes that are both part of either the antecedence or consequence pattern.

Fig. 5A shows an example of a start-start constraint on the execution sequence of tasks. It depicts the process perspective of compliance rule c_2 from Tab. 1. Note that this visual compliance rule disallows executing task *change order* after the start of task *production*.

4.3 Interaction Perspective

The interaction perspective covers constraints on the messages exchanged and the *interaction view* of the eCRG meta-model. Message exchanges are expressed in terms of particular nodes that reflect the events of *sending* and *receiving a message*. In turn, a *message flow* denotes the dependency between the events representing the sending and receiving of a particular message (cf. Fig. 4).

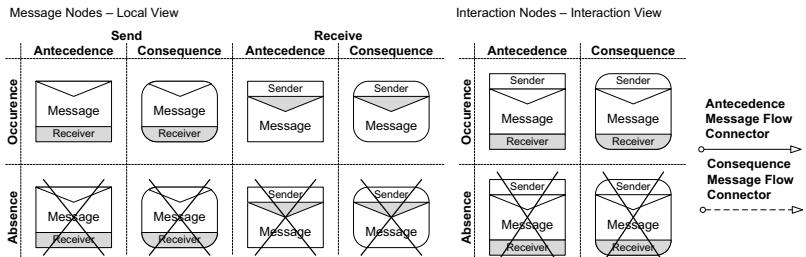


Fig. 4. eCRG elements of the interaction perspective

In Fig. 5B, the elements from Fig. 4 are used to model the process and interaction perspective of compliance rule c_4 . This rule requires that after receiving message *request* from a reseller, a *solvency check* must be performed first. Then, a decision about *approval* has to be made before replying the request. Although the rule modeled in Fig. 5B considers the interaction perspective, using the two message nodes *request* and *reply*, it still represents the view of a particular business partner on its local business processes. We refer to this traditional point of view as the *local view* of a compliance rule. However, when considering the choreography diagram of BPMN 2.0 or compliance rules c_1 and c_3 from Tab. 1, one can easily discover a global point of view on cross-organizational processes and related interactions (i.e., the messages exchanged). In this *interaction view*, interaction nodes (cf. Fig. 4) are used to denote the exchange of a message between two business partners. Since the interaction view spans multiple business partners, task nodes may be annotated with the executing business partner if required (cf. Fig. 3).

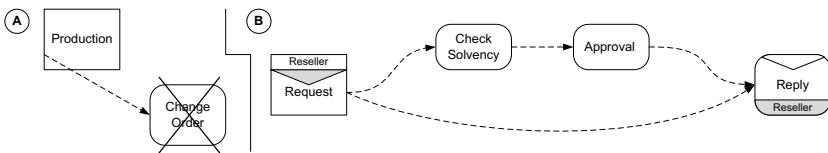


Fig. 5. Local view on c_2 and c_4 with process and interaction perspectives

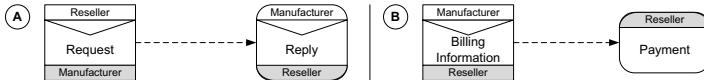


Fig. 6. Interaction view on c_1 and c_3 with process and interaction perspectives

Fig. 6A provides an interaction view on compliance rule c_1 from Tab. 1: After the reseller sends a *request* to the manufacturer, eventually, the manufacturer must *reply*. Further, Fig. 6B provides an interaction view on compliance rule c_3 from Tab. 1. This rule requires that the reseller must perform task *payment* after having received *billing information* from the manufacturer.

4.4 Time Perspective

Having a closer look on the original definition of compliance rules c_1 and c_3 from Tab. 1, it becomes clear that Figs. 6A and 6B do not fully cover them yet. In particular, the distance in time between the interactions and tasks have not been considered. Fig. 7 provides elements for modeling *points in time* and *time conditions* in compliance rules. The latter may be attached to task nodes as well as sequence or message flow connectors to either constrain the duration of a task or the time distance between tasks, messages, and points in time. Additionally, *time distance connectors* are introduced that must be attached with a time condition. Respective time distance connectors and related time conditions then

Points in Time	Anteced.	Consequ.	Particular Point in Time	Time Conditions	Anteced. Sequence Distance Connector
Occurrence					
Absence					

Fig. 7. eCRG elements of the time perspective

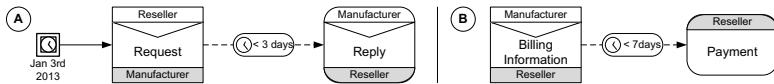


Fig. 8. Interaction view on c_1 and c_3 with process, interaction, and time perspectives

allow constraining the time distance between tasks, messages, and points in time without implying a particular sequence.

Fig. 8A combines the interaction and time perspectives of compliance rule c_1 . This visual representation of c_1 covers exactly the semantics of the compliance rule described in Tab. 1. In Fig. 8B, the interaction and time perspectives of c_3 are provided. This compliance rule requires that at most seven days after the manufacturer sends *billing information* to the reseller, the latter must perform task *payment*.

4.5 Resource Perspective

The resource perspective covers the different kinds of human resources as well as their inter-relations, and it allows constraining the assignment of resources to tasks. In particular, we consider resources like *staff member*, *role*, *group*, and *organizational unit*, and their relation to tasks. Furthermore, we support *resource conditions* and *relations* among resources (cf. Fig. 9). Similar to task nodes, *resource nodes* may be part of the antecedence or consequence pattern. Alternatively, they may represent a particular resource instance (e.g. staff member *Mr. Smith*, or role *supervisor*). In turn, *resource conditions* may constrain resource nodes. Further, the *performing relation* indicates the performer of a task. Finally, *resource relation connectors* express relations between resources. Note that the resource perspective can be easily extended with other kinds of resources if required.

Fig. 10 combines the process, interaction, time, and resource perspectives of compliance rule c_4 . This rule requires that at least three days after receiving a *request* of the reseller, a *reply* must be sent to him. Before sending this reply, first of all, task *solvency check* must be performed by a staff member assigned to the particular organizational unit *order processing department*. Following this task, another staff member of the same department with *supervisor* status (i.e., role) must decide whether or not to grant approval before sending the *reply*.

	Data Nodes	Resource Nodes	Data/Resource Conditions
Antecedence	Data Container Data Object	Group Staff Member Role Organizational Unit	Antecedence > value Consequence < value
Consequence	Data Container Data Object	Group Staff Member Role Organizational Unit	Antecedence Data Flow Connector Consequence Data Flow Connector
Particular Resources	Requests Request from Mrs Ly	Quality Managers Mr Knup Computer Scientist Dept. Information Systems	Antecedence Performing and Relation Connector Consequence Performing and Relation Connector

Fig. 9. eCRG elements of the resource and data perspectives

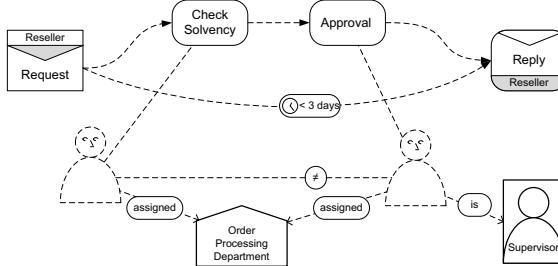


Fig. 10. Local view on c_4 with process, interaction, time, and resource perspectives

4.6 Data Perspective

Fig. 9 introduces elements for modeling data containers and data objects as well as connectors representing data flow. Thereby, *data containers* refer to process data elements or global data stores. By contrast, *data objects* refer to particular data values and object instances. Similar to resource nodes, *data nodes* may be part of the antecedence or consequence pattern, or represent a particular data container or data object (e.g., data container *student credit points*, document *1st order from Mr. Smith*). Further, *data flow* defines which process tasks read or write which data objects or data container. To constrain data container, data objects, and data flow, *data conditions* may be attached. Finally, *data relation connectors* may be used either to compare different data objects or to constrain the value of data containers at particular points in time.

Figs. 11A, 11B, and 11C show the visual modeling of compliance rules c_2 , c_3 , and c_4 covering the data perspective as well as the other perspectives discussed. Each of the depicted eCRGs covers the informal semantics described in Tab. 1.

5 Discussion and Validation

Sect. 4 introduced the eCRG language, which comprises various elements for modeling the process, interaction, time, resource, and data perspectives of

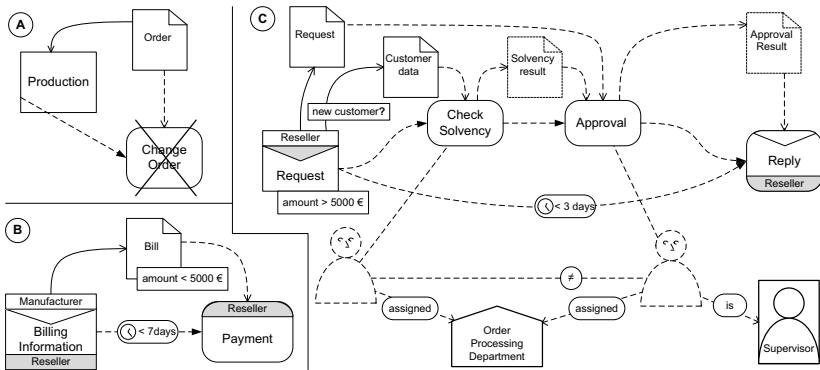


Fig. 11. Local view on c_2 and interaction view on c_3 and c_4 , considering process, interaction, time, resource, and data perspectives

compliance rules. However, note that the introduced elements must not be arbitrarily combined, but should follow syntactic constraints. First, any eCRG must be acyclic. Second, antecedence and consequence connectors must be applied in a reasonable way, e.g., any sequence flow between an antecedence absence and a consequence absence node does not make sense, and hence is forbidden. Third, the use of attachments is restricted in a similar way. Finally, exclusive and alternative connectors must only connect tasks, messages, or interaction nodes of the same pattern. Fig. 13 summarizes valid and invalid use cases of connectors and attachments.

To the best of our knowledge, our approach is the first one that allows modeling compliance rules visually considering the interaction, time, resource, and data perspectives. Note that there exist pattern-based approaches that model compliance rules supporting at least the time, resource, and data perspectives [6, 8, 34]. These patterns resulted from literature and case studies, and thus constitute a suitable empirical basis for evaluating the appropriateness of our approach. Therefore, we modeled the compliance patterns introduced in [6, 8, 34] with our visual notation in [35]. Overall, we were able to fully model 26 out of the 27 business process control patterns from [8], including 5 time patterns and 7 resource patterns as well. Only the multi-segregation pattern cannot be modeled as eCRG [35]. Further, eCRGs allow modeling each of the 15 control flow compliance rules as well as the data flow restrictions and organizational aspects (i.e., separation of duty) from [6]. Finally, the time-patterns introduced in [16] can also be covered with eCRGs [35]. Note that the proposed visual notation (i.e., eCRG) is not restricted to these patterns (e.g., c_4 cannot be modeled by the use of compliance patterns).

Any syntactically correct eCRG can be converted into a corresponding FOL formula. The FOL formula, in turn, can be evaluated over process traces, including the process, interaction, time, resource, and data perspectives. Furthermore, the internal consistency (i.e., absence of conflicts) of a set of compliance rules can be verified. We provide details on the transformation of eCRG into FOL

Turetken et al.			Ramezani et al.			Lanz et al.		
Precedes	++	USegregatedFrom	++	Existence	++	Time Lags between Activities	++	
LeadsTo	++	BondedWith	++	Bounded Existence	+	Durations	++	
XLeadsTo	+	RBondedWith	++	Bounded Sequence	+	Time Lags between Events	++	
PLeadsTo	++	Multi-Segregated	-	Parallel	++	Fixed Date Elements	++	
ChainLeadsTo	++	Multi-Bonded	++	Precedence	++	Schedule Restricted Elements	++	
Chain Precedes	++	Within k	++	Chain Precedence	++	Time-based Restrictions	+	
LeadsTo - Else	++	After k	++	Response	++	Validity Period	++	
Exists	++	ExactlyAt k	++	Chain Response	++	Time-dependend Variability	+	
Absent	++	Exists Max/Min	+	Between	+	Cyclic Elements	++	
Universal	+	Exists Every k	+	Exclusive	++	Periodicity	++	
CoExists	++			Mutual Exclusive	++			
CoAbsent	++			Inclusive	++			
Exclusive	++			Prerequisite	++			
CoRequisite	++			Substitute	++			
MutexChoice	++			Corequisite	++			
PerformedBy	++			Restricted data values	++			
SegregatedFrom	++			Separation of Duty	++			

++ full support, + inconvenient support, 0 partial support, - minor support, -- no support

Fig. 12. Support of compliance patterns with eCRGs

antecedence connector sequence flow		consequence connector sequence flow				antecedence attachments		exclusive / alternative connectors			
		AO	AA	CO	CA	AO	AA	CO	CA	AO	AA
AO	ok	ok		AO	ok	X	ok	ok		AO	ok
AA	ok	X		AA	X	X	X	X		AA	ok
CO				CO	ok	X	ok	ok		CO	X
CA				CA	ok	X	ok	X		CA	X

AO antecedence occurrence node
AA antecedence absence node
CO consequence occurrence node
CA consequence absence node

Fig. 13. Valid Use of eCRG elements

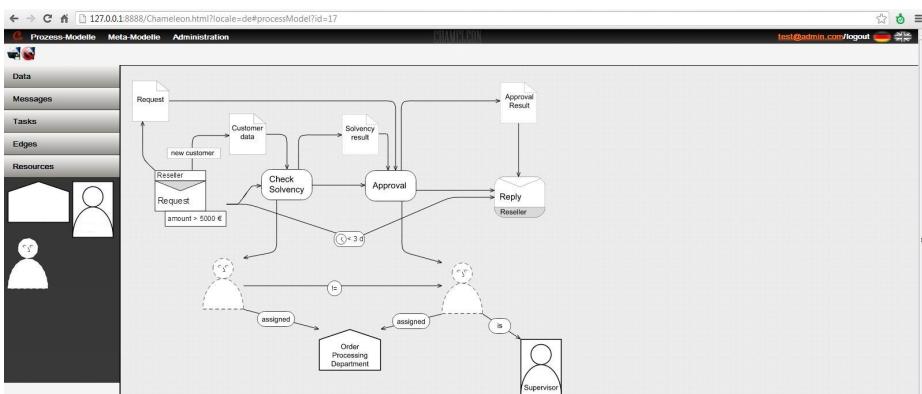


Fig. 14. Proof-of-concept implementation

formula (i.e., the formal semantics of eCRGs) and the subsequent verification over process traces in [35]. We have demonstrated the feasibility of our modeling approach by implementing a proof-of-concept prototype of a modeling environment for eCRGs, which we then used to model compliance rules from a variety of domains including higher education, medicine and automotive engineering. In particular, domain experts have been able to define and understand the visual notation we used. We are currently preparing user experiments to check how end users deal with large sets of visual compliance rules. Fig. 14 provides a screenshot of our prototype.

6 Summary and Outlook

While compliance rule modeling has been introduced by a plethora of approaches, the data, time, and resource perspectives of compliance rules have not been sufficiently addressed yet [5–8]. This paper introduces extensions for visual compliance rule languages to support these perspectives. In particular these extensions are introduced as part of *extended compliance rule graphs (eCRG)*, which are based on the *compliance rule graph (CRG) language* [10, 12]. However, the modeling elements described may be applied to other compliance rule languages as well. Besides the data, time, and resource perspectives, we further suggest elements for modeling the interaction perspective of compliance rules. To provide tool support for both the modeling and verification of compliance rules, we formalize the syntax and semantics of eCRGs in a technical report [35]. Finally, pattern-based analysis has shown that eCRGs have sufficient expressiveness. Our next step will be an experiment to evaluate the usability and scalability of eCRGs. Further, we will apply the proposed extensions to other compliance rule languages. Finally, we will develop techniques for verifying compliance of business processes and process choreographies with such rules.

References

1. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems. Springer (2012)
2. van der Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997)
3. Sadiq, W., Governatori, G., Namiri, K.: Modeling control objectives for business process compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 149–164. Springer, Heidelberg (2007)
4. Knuplesch, D., Reichert, M., Mangler, J., Rinderle-Ma, S., Fdhila, W.: Towards compliance of cross-organizational processes and their changes. In: La Rosa, M., Soffer, P. (eds.) BPM 2012 Workshops. LNBP, vol. 132, pp. 649–661. Springer, Heidelberg (2013)
5. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: Hints on how to face business process compliance. In: JISBD 2010 (2010)
6. Ramezani, E., Fahland, D., van der Aalst, W.M.P.: Where Did I Misbehave? Diagnostic Information in Compliance Checking. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 262–278. Springer, Heidelberg (2012)

7. Mangler, J., Rinderle-Ma, S.: Iupc: Identification and unification of process constraints. arXiv.org (2011)
8. Turetken, O., Elgammal, A., van den Heuvel, W.J., Papazoglou, M.: Capturing compliance requirements: A pattern-based approach. IEEE Soft., 29–36 (2012)
9. Ghose, A.K., Koliadis, G.: Auditing business process compliance. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 169–180. Springer, Heidelberg (2007)
10. Ly, L.T., Rinderle-Ma, S., Dadam, P.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 9–23. Springer, Heidelberg (2010)
11. Liu, Y., Müller, S., Xu, K.: A static compliance-checking framework for business process models. IBM Systems Journal 46(2), 261–335 (2007)
12. Knuplesch, D., Reichert, M.: Ensuring business process compliance along the process life cycle. Technical Report 2011-06, Ulm University (2011)
13. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow resource patterns: Identification, representation and tool support. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 216–232. Springer, Heidelberg (2005)
14. Wang, J., Kumar, A.: A framework for document-driven workflow systems. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 285–301. Springer, Heidelberg (2005)
15. Eder, J., Tahamtan, A.: Temporal conformance of federated choreographies. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 668–675. Springer, Heidelberg (2008)
16. Lanz, A., Weber, B., Reichert, M.: Time patterns for process-aware information systems. Requirements Engineering (2012)
17. Decker, G., Weske, M.: Interaction-centric modeling of process choreographies. Inf. Sys. 35(8) (2010)
18. Barros, A., Dumas, M., ter Hofstede, A.H.M.: Service interaction patterns. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 302–318. Springer, Heidelberg (2005)
19. Knuplesch, D., et al.: Data-aware interaction in distributed and collaborative workflows: Modeling, semantics, correctness. In: CollaborateCom 2012, pp. 223–232 (2012)
20. Ly, L.T., et al.: Integration and verification of semantic constraints in adaptive process management systems. Data & Knowl. Eng. 64(1), 3–23 (2008)
21. Ly, L.T., et al.: On enabling integrated process compliance with semantic constraints in process management systems. Inf. Sys. Front. 14(2), 195–219 (2012)
22. Ramezani, E., Fahland, D., van der Werf, J.M., Mattheis, P.: Separating compliance management and business process management. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part II. LNBP, vol. 100, pp. 459–464. Springer, Heidelberg (2012)
23. Governatori, G., Sadiq, S.: The journey to business process compliance. In: Handbook of Research on BPM, pp. 426–454. IGI Global (2009)
24. Namiri, K., Stojanovic, N.: Pattern-Based design and validation of business process compliance. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 59–76. Springer, Heidelberg (2007)
25. Governatori, G., Hoffmann, J., Sadiq, S., Weber, I.: Detecting regulatory compliance for business process models through semantic annotations. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008 Workshops. LNBP, vol. 17, pp. 5–17. Springer, Heidelberg (2009)

26. Kumar, A., Yao, W., Chu, C.: Flexible process compliance with semantic constraints using mixed-integer programming. *INFORMS J. on Comp.* (2012)
27. Awad, A., Weidlich, M., Weske, M.: Specification, verification and explanation of violation for data aware compliance rules. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC-ServiceWave 2009. LNCS, vol. 5900, pp. 500–515. Springer, Heidelberg (2009)
28. Knuplesch, D., Ly, L.T., Rinderle-Ma, S., Pfeifer, H., Dadam, P.: On enabling data-aware compliance checking of business process models. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 332–346. Springer, Heidelberg (2010)
29. Kokash, N., Krause, C., de Vink, E.: Time and data aware analysis of graphical service models. In: SEFM 2010 (2010)
30. Höhn, S.: Model-based reasoning on the achievement of business goals. In: SAC 2009, pp. 1589–1593. ACM, New York (2009)
31. Accorsi, R., Lowis, L., Sato, Y.: Automated certification for compliant cloud-based business processes. *Business & Inf. Sys. Engineering* 3(3), 145–154 (2011)
32. Awad, A., Weidlich, M., Weske, M.: Visually specifying compliance rules and explaining their violations for business processes. *Vis. Lang. Comp.* 22(1), 30–55 (2011)
33. Feja, S., Speck, A., Witt, S., Schulz, M.: Checkable graphical business process representation. In: Catania, B., Ivanović, M., Thalheim, B. (eds.) ADBIS 2010. LNCS, vol. 6295, pp. 176–189. Springer, Heidelberg (2010)
34. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Property specification patterns for finite-state verification. In: FMSP 1998 (1998)
35. Knuplesch, D., et al.: On the formal semantics of the extended compliance rule graph. Technical Report 2013-05, Ulm University (2013)

Deciding Data Object Relevance for Business Process Model Abstraction

Josefine Harzmann, Andreas Meyer, and Mathias Weske

Hasso Plattner Institute at the University of Potsdam

`Josefine.Harzmann@student.hpi.uni-potsdam.de,`
`{Andreas.Meyer,Mathias.Weske}@hpi.uni-potsdam.de`

Abstract. Business process model abstraction received considerable attention lately. So far, business process model abstraction is mainly based on control flow aspects ignoring data objects. Recently, data objects have been included in process model abstraction techniques. Thereby, the question arises which data objects are relevant for a specific abstraction level of a process model. To date, data object relevance is decided by control flow. But for data object abstraction independently from control flow aspects, the relevance question remains open. In this paper, we answer this question by introducing a set of data object relevance criteria focusing on data objects. These have been derived from use cases and have been evaluated with a first user study. Further, we show means to combine the presented criteria.

Keywords: BPM, Data Objects, Relevance Criteria, Data Object Abstraction, Business Process Model Abstraction.

1 Introduction

Business process management is an approach to systematically organize the operations occurring in an organization's daily business. They are described by the means of process models, which contain partially ordered activities describing actual work items to be performed during the operations. Furthermore, process models contain role and data object information to determine who actually executes an activity and which data objects are utilized, i.e., read or written by the activity [12]. Hence, process models contain much information. But not every stakeholder is interested in all such details. Process model abstraction is a technique to provide different views on a model. Information not relevant for a stakeholder is suppressed or aggregated. For example, a manager may want to know about the main activities and their behavior, whereas a process analyst is interested in time-consuming and cost-intensive activities to identify optimization opportunities. To determine the relevant information for a stakeholder, abstraction criteria are utilized. Current process model abstraction approaches mainly focus on control flow aspects as, for instance, activities [1, 2, 8–10], but recently also data objects were considered as extension to control flow abstraction [5]. However, many use cases exist where control flow shall be kept as is while the view on data objects shall be restricted or adjusted.

In this paper, we introduce a set of data object relevance criteria extracted from several use cases. These criteria focus on data objects utilizing the information given in the process models (activity execution order) and known from process statistics (branch

probability) to decide for each data object whether it is relevant or not. In the field of process model abstraction, these criteria can be utilized for abstraction of data objects.

The remainder of the paper is structured as follows. Section 2 introduces preliminaries required for all data object relevance criteria, which we describe one by one followed by means to combine several of them and a sketch of our evaluation in Section 3. Section 4 discusses related work and Section 5 concludes the paper.

2 Preliminaries

The relevance criteria base on various notions, which we briefly introduce in this section.

A *data object* is an entity being processed, manipulated, or worked with during process execution. Each object contains a *data state*, which describes a specific business situation from the data object's point of view. The data state changes during process execution. The dependencies between all objects potentially be used in a process model are presented in a *data model* represented as subset of an UML class diagram [7]. The set of all data objects is denoted with D . Each object $d \in D$ refers to a unique data object class $dc \in DC$ and can be mapped to it by function $\varphi : D \rightarrow DC$ such that $\varphi(d) \in DC$. The dependencies are presented with relations $\mathfrak{R} \subseteq DC \times DC$ of types association (\mathfrak{R}_{Assoc}), aggregation (\mathfrak{R}_{Aggr}), composition (\mathfrak{R}_{Comp}), and generalization (\mathfrak{R}_{Gen}) [5].

A *process model* $M = (N, D, R, C, DF, \alpha, \beta, \gamma, \kappa, \lambda)$ consists of a finite non-empty set $N \subseteq A \cup G$ of nodes comprising activities A and gateways G , a finite non-empty set D of data objects, and a finite non-empty set R of roles executing a given process. $C \subseteq N \times N$ is the control flow relation defining a partial ordering of nodes and $DF \subseteq (A \times D) \cup (D \times A)$ is the data flow relation defining write and read access to objects. It can be generalized by function $\alpha : A \times D \rightarrow \{0, 1\}$, which evaluates to 0, if object $d \in D$ is not accessed by activity $a \in A$, i.e., $(a, d) \notin DF$ and $(d, a) \notin DF$, and which evaluates to 1, if a read or write association exists. Function $\beta : A \rightarrow R$ assigns to each activity a role, which executes the corresponding activity. Function $\gamma : D \rightarrow 2^R$ assigns to each object a set of roles specifying who is allowed to access the corresponding object; 2^R denotes the power set of R . Let $a \in A$ be an activity, $d \in D$ be a data object, and $r \in R$ be a role, then d is accessed by r during execution of a , if $\beta(a) \in \gamma(d)$ and $\alpha(a, d) = 1$. $\kappa : C \rightarrow (0, 1]$ defines the probability a control flow edge is chosen for execution. The probabilities for all outgoing edges of a node must sum up to one. Function $\lambda : D \times A \rightarrow (0, 1]$ determines the probability a specific object $d_1 \in D$ is accessed by an activity $a \in A$. λ evaluates to 1, if there does not exist another object $d_2 \in D$ of the same class, i.e., $\varphi(d_1) \neq \varphi(d_2)$, read (written) by a . If several objects d_i of the same class, i.e., let $dc \in DC$ be a data object class then $\forall d_i \in D : \varphi(d_i) = dc$, are read (written) by a , the probability is determined by the execution probability of preceding (succeeding) branches, where the corresponding objects are accessed. The probabilities of data objects of one class read (written) by one activity has to sum up to one. Let $a \in A$ be an activity and $dc \in DC$ a data object class then $\forall d_i \in D$ such that $\varphi(d_i) = dc \wedge (a, d_i) \in F : \sum_{d_i} \lambda(d_i) = 1$.

Utilizing the weak order relation from behavioral profiles [3, 11], the order of two activities is defined as $> \subseteq A \times A$. Let $a_1, a_2 \in A$ be two activities then $a_1 > a_2$ means that there exists a path in the process model such that a_2 is executed after a_1 .

3 Data Object Relevance Criteria

In this section, we introduce a set of eleven data object relevance criteria clustered into three groups: (i) Binary criteria, (ii) Continuous criteria, and (iii) Hierarchical criteria. Binary criteria can be seen as filters – objects matching the filter are considered relevant while the others are not. In contrast, continuous criteria require the user to define a threshold value and an equality condition concerning this value. If an object satisfies the condition, it is considered relevant. The hierarchical criteria are based on the hierarchical level of data object classes defined in the data model and allows the combination of several objects to a single one along the class diagram dependencies. Indeed, there do exist data object relevance criteria, which we did not add explicitly to our criteria set. But the additional ones we identified can be described by combining criteria that are part of the set introduced in this paper. To decide data object relevance, we define the relevance function $rel_k(d)$, which evaluates for a given object $d \in D$ to 1 or 0 for criteria from group one or two ($rel_{i,ii} : D \rightarrow \{0, 1\}$) and which evaluates to an object $e \in D$ for hierarchical data object relevance criteria for a given object $d \in D$ based on the set of objects to be considered ($rel_{iii} : D \rightarrow D$). Following, the relevance function is defined as $rel_k : D \rightarrow \{0, 1\} \cup D$.

Next, we introduce the data object relevance criteria one by one. For each criterion, we informally describe the criterion and we provide a formal representation utilizing the relevance function $rel_k(d)$. The criteria base on use cases derived from initial discussions with process experts and from process models describing scenarios in different domains.

3.1 Binary Criteria

We start with the introduction of the five binary data object relevance criteria.

B1 – Filter for Data Object Class. The relevance function rel_1 bases on the class of a data object and can be derived directly from the selection of the stakeholder; $DC_{sel} \subseteq DC$ comprises these classes. For a given object $d \in D$, the relevance function evaluates to 1, if the corresponding class $dc \in DC$ is chosen, i.e., $dc \in DC_{sel}$, and it evaluates to 0, if the data object class is not chosen by the stakeholder.

$$rel_1(d) = \begin{cases} 0 & \text{if } \varphi(d) \notin DC_{sel} \\ 1 & \text{if } \varphi(d) \in DC_{sel} \end{cases} \quad (1)$$

B2 – Filter for Type of Access. Each data object access is directed indicating either a read access $D \times A$ or a write access $A \times D$. An activity can read (write) a data object several times if there are alternative input (output) states. The stakeholder selects either type of access $t \in \{read, write\}$ resulting in a representation of read (write) accesses only. The access to a data object $d \in D$ satisfies a selection t for an activity $a \in A$ ($\alpha(a, d) = 1$), if $(d, a) \in DF$ for a read access or $(a, d) \in DF$ for a write access. For a given object d , the relevance function rel_2 evaluates to 1 or 0 accordingly to the satisfaction of selection t . All accesses to a data object not being of type t are irrelevant.

$$rel_2(d) = \begin{cases} 0 & \text{if } (t = read \wedge (d, a) \notin DF) \vee (t = write \wedge (a, d) \notin DF) \\ 1 & \text{if } (t = read \wedge (d, a) \in DF) \vee (t = write \wedge (a, d) \in DF) \end{cases} \quad (2)$$

B3 – Filter for First Access. Using the weak order relation $>$, for a given object $d \in D$, the relevance function rel_3 evaluates to 1, if no object belonging to the same class as d is accessed before d on any path of the process model, and it evaluates to 0, if there exists an object belonging to the same class as d , which is accessed before d on some path.

Note that more than one data object belonging to the same class can be considered relevant with this criterion. This is the case for alternative accesses, i.e., several reads or writes of one object in different states by one activity, for parallel branches, and for exclusive branches. When an activity $a \in A$ reads and writes data objects of the same data object class, the reading access is executed before the writing access such that the read objects may be considered relevant and the written ones are not.

$$rel_3(d) = \begin{cases} 0 & \text{if } \exists a_1, a_2 \in A, \alpha(a_1, d) = 1, a_2 > a_1 : \\ & \exists e \in D, \varphi(e) = \varphi(d), \alpha(a_2, e) = 1 \\ 1 & \text{if } \exists a \in A, \alpha(a, d) = 1 : \forall a_i \in A, a_i > a : \\ & \forall d_j \in D, \varphi(d_j) = \varphi(d), \alpha(a_i, d_j) = 0 \end{cases} \quad (3)$$

B4 – Filter for Last Access. Analogously to B3, we use the weak order relation $>$. For a given object d , the relevance function rel_4 evaluates to 1, if no object belonging to the same class as d is accessed after d on any path of the process model, and it evaluates to 0, if there exists an object belonging to the same class as d , which is accessed after d on some path. Analogously to criterion B3, several data objects belonging to the same data object class may be considered relevant with this criterion.

$$rel_4(d) = \begin{cases} 0 & \text{if } \exists a_1, a_2 \in A, \alpha(a_1, d) = 1, a_1 > a_2 : \\ & \exists e \in D, \varphi(e) = \varphi(d), \alpha(a_2, e) = 1 \\ 1 & \text{if } \exists a \in A, \alpha(a, d) = 1 : \forall a_i \in A, a > a_i : \\ & \forall d_j \in D, \varphi(d_j) = \varphi(d), \alpha(a_i, d_j) = 0 \end{cases} \quad (4)$$

B5 – Filter for Execution Role. The relevance function rel_5 bases on the set of roles $R_{sel} \subseteq R$ chosen by the stakeholder. Given an object $d \in D$, the relevance function evaluates to 1, if the role $r \in R$ accessing d during execution of activity $a \in A$ is selected, i.e., $r = \beta(a) \in \gamma(d) \wedge \alpha(a, d) = 1 \wedge r \in R_{sel}$, and it evaluates to 0, if it is not selected.

$$rel_5(d) = \begin{cases} 0 & \text{if } \forall a_i \in A : \alpha(a_i, d) = 0 \vee \beta(a_i) \notin \gamma(d) \cap R_{sel} \\ 1 & \text{if } \exists a \in A : \alpha(a, d) = 1 \wedge \beta(a) \in \gamma(d) \cap R_{sel} \end{cases} \quad (5)$$

3.2 Continuous Criteria

Next, we will introduce the continuous relevance criteria, which utilize user defined threshold values on an ordinal scale to decide relevance. For that threshold value, the user needs to specify an equality condition $EC = \{=, <, >, \leq, \geq\}$. Given a data object $d \in D$, a criterion-specific relevance indicator ri , the threshold value v , and an equality

condition $ec \in EC$, the relevance function rel_6 evaluates to 1, if the equation is true, and it evaluates to 0, if the equation is false.

$$rel_6(d) = \begin{cases} 0 & ri \times ec \times v \text{ is false} \\ 1 & ri \times ec \times v \text{ is true} \end{cases} \quad (6)$$

Thereby, the interval for valid threshold values depends on the relevance criterion; also the type may differ, as for instance, indicated by percent and natural numbers.

C1 – Access Probability. Applying this criterion requires the stakeholder to specify the threshold value in percent. The relevance indicator is the probability function $\omega_1 : D \rightarrow (0, 1]$ mapping a data object to a value corresponding to the range from 0% to 100%. ω_1 is determined by the data object access probability and the execution probability of activity $a \in A$ that accesses object $d \in D$ (cf. λ, κ in Section 2).

$$\omega_1(d) = \lambda(d, a) \cdot \kappa(a) \text{ with } \alpha(a, d) = 1 \quad (7)$$

C2 – Absolute Number of Occurrences. For applying this criterion, the stakeholder has to specify the threshold value, a positive natural number. The relevance indicator $\omega_2 : DC \rightarrow \mathbb{N}^+$ maps a data object class to a positive natural number counting distinct activities accessing object $d \in D$ of the same class $dc \in DC$. When an activity reads (writes) two alternative objects of the same class, it is counted as one access. If objects are accessed within a loop structure, each object access is counted once independently of the actual number of loop executions aligning with the size computation of process models as metric [4] or of Petri nets used to model concurrent systems [6].

$$\omega_2(dc) = \sum_{a_i \in A} \sum_{\varphi(d_j) = dc} (\alpha(a_i, d_j) \cdot \lambda(d_j, a_i)) \quad (8)$$

C3 – Relative Number of Occurrences. Applying this criterion requires the stakeholder to specify the threshold value in percent. The relevance indicator $\omega_3 : DC \rightarrow (0, 1]$ maps an object to a value corresponding to the range from 0% to 100%. For a given data object class $dc \in DC$, it can be directly derived from the absolute number of occurrences ω_2 by dividing that result by the sum of all object accesses. The remarks with respect to alternative accesses and loop structures apply as described above for C2.

$$\omega_3(dc) = \frac{\omega_2(dc)}{\sum_{dc_i \in DC} \omega_2(dc_i)} \quad (9)$$

C4 – Weighted Number of Occurrences. For application, the stakeholder has to specify the threshold value as a positive real number. The relevance indicator is the function $\omega_4 : DC \rightarrow \mathbb{R}^+$ mapping a data object class to a positive real number by summing up the access probabilities for all objects $d_i \in D$ belonging to one class $dc \in DC$ utilizing probability function ω_1 .

$$\omega_4(dc) = \sum_{\varphi(d_i) = dc} \omega_1(d_i) \quad (10)$$

3.3 Hierarchical Criteria

The combination, i.e., composition or generalization, of data objects may happen either to the root node of or to the least common denominator in the data model accompanying the process model. For each relation type except association, we extract a *tree* from the data model based on which the root or the least common denominator is identified for a given data object class considering all data object classes utilized in a process model or by a set of activities – depending on the stakeholder’s criterion configuration.

A data object class $dc_1 \in DC$ is a superclass of class $dc_2 \in DC$, if dc_1 is a parent of dc_2 in the tree extracted from the data model. Thereby, dc_1 is more generic than dc_2 for a generalization relation and dc_1 contains dc_2 for an aggregation or composition relation. Formally, we define such superclass relation between classes dc_1 and dc_2 by $(dc_1, dc_2) \in \mathfrak{R}_k$ with k representing the specific relation type and \mathfrak{R}_k containing the relations to be considered for criteria application, e.g., only relations between classes accessed by a specific activity or all classes utilized in the process model.

H1 – Combination to Root Node. For applying this criterion, the stakeholder has to specify the set of relations \mathfrak{R}_k . For a given data object d , the relevance function rel_7 evaluates to the input data object, if there does not exist a superclass for the corresponding class $dc = \varphi(d)$, and it evaluates to data object $\varphi^{-1}(dc)$ of class $dc \in DC$, if dc is a superclass of $\varphi(d)$ and if dc is the root node of the respective tree extracted from the data model. The data state of the resulting data object needs to be defined by the stakeholder.

$$rel_7(d) = \begin{cases} d & \text{if } \nexists dc \in DC : (dc, \varphi(d)) \in \mathfrak{R}_k \\ \varphi^{-1}(dc_i) & \text{if } \exists dc_i \in DC : (dc_i, \varphi(d)) \in \mathfrak{R}_k \\ & \wedge \nexists dc_j \in DC : (dc_j, dc_i) \in \mathfrak{R}_k \end{cases} \quad (11)$$

H2 – Combination to Least Common Denominator. The least common denominator $LCD \in DC$ of a set of data object classes $X \subseteq DC$ is determined by function $\zeta : 2^X \rightarrow X$ adapted from [5]. For applying this criterion, the stakeholder has to specify the set of relations \mathfrak{R}_k . For a given data object d , the relevance function rel_8 evaluates to the input data object, if there does not exist a superclass for the corresponding class $dc = \varphi(d)$ or if dc is the least common denominator, and it evaluates to data object $\varphi^{-1}(dc)$ of class $dc \in DC$, if dc is a superclass of $\varphi(d)$ and if dc is the least common denominator. The data state of the resulting data object needs to be defined by the stakeholder.

$$rel_8(d) = \begin{cases} d & \text{if } (\nexists dc \in X : (dc, \varphi(d)) \in \mathfrak{R}_k) \vee (\varphi(d) = \zeta(X)) \\ \varphi^{-1}(dc_i) & \text{if } \exists dc \in X : (dc, \varphi(d)) \in \mathfrak{R}_k \wedge dc = \zeta(X) \end{cases} \quad (12)$$

3.4 Combination of Relevance Criteria

After introducing all relevance criteria one by one, we will proceed with means to combine several of these atomic criteria resulting in one complex criterion. For instance, the stakeholder is interested in all data objects that are written at least three times by a

specific role (combining criteria B2, B5, and C2). To define those complex relevance criteria, the stakeholder defines a schema \mathcal{S} , which allows to combine the introduced atomic criteria in three different ways: sequentially \mapsto , in conjunction \cup , and in intersection \cap . Thereby, the schema may contain several types of combination for different relevance criteria. Next, we will introduce all combination types. Although we will use two criteria for the formalization of these types, any number of relevance criteria can be combined the described ways. Thereby, a complex criterion is considered atomic such that it may also be combined with other relevance criteria.

The relevance function $rel_{\Sigma} : D \rightarrow \{0, 1\} \cup D$ indicates the combined relevance. We formally define the sequential execution using stacked functions. Given two relevance criteria rc_1, rc_2 , an object $d \in D$, and a combination schema stating that rc_1 shall be executed before rc_2 , the relevance function $rel_{\Sigma}(d)$ is computed by first applying the relevance function rel_{rc_1} corresponding to rc_1 on object d and then applying relevance function rel_{rc_2} corresponding to rc_2 on the result of $rel_{rc_1}(d)$. If an inner function evaluates to 1, object d is used as input to the directly surrounding function ($rel_{rc_k}(d) = 1 \Rightarrow d$). If an inner function evaluates to 0, the empty set is used as input to the surrounding function ($rel_{rc_k}(d) = 0 \Rightarrow \emptyset$). A function having the empty set as input evaluates to 0 ($rel_{rc_k}(\emptyset) = 0$).

$$rel_{\Sigma}(d) = rel_{rc_2}(rel_{rc_1}(d)) \quad (13)$$

In Equation 13, rel_{rc_1} is an inner function and rel_{rc_2} is the outer function as it is not surrounded by any other function. If $rel_{rc_1}(d)$ evaluates to 1, then d is used as input to rel_{rc_2} . Otherwise, the empty set is used.

The combination of data object relevance criteria can be symmetric, i.e., the result is independent from the criteria order, or asymmetric. If a schema contains one of the following criteria, it is asymmetric: B3, B4, or any continuous or hierarchical criterion.

While the sequential execution can be applied to all relevance criteria, the two remaining ones can only be applied to binary and continuous criteria. For both types, the combination bases on executing all defined relevance criteria separately and logically combining the results with *or* (conjunction) or *and* (intersection) operators.

$$\begin{aligned} \text{conjunction: } & rel_{\Sigma}(d) = rel_{rc_1}(d) \vee rel_{rc_2}(d) \\ \text{intersection: } & rel_{\Sigma}(d) = rel_{rc_1}(d) \wedge rel_{rc_2}(d) \end{aligned} \quad (14)$$

A schema returning the objects being accessed at least twice by a certain role, which are also the first or last write access, looks as follows: $\mathcal{S} = (B5 \cap C2) \mapsto B2 \mapsto (B3 \cup B4)$.

3.5 Evaluation

We conducted two experiments. First, we asked eight participants from academia with a strong background in business process management to derive relevance criteria from given process models acting in different roles for each model. We then compared them to the criteria we introduced in this paper. Altogether, except for criteria B1 and B5 (mentioned in 40% of all use cases) and criteria C1 and C4 (not mentioned at all), the criteria highly depend on the use case, because the participants answered homogeneously

considering similar criteria for the same question in the questionnaire. Secondly, we provided eleven process experts with the introduced criteria and asked for their rating on a Likert scale from 1 (excellent) to 6 (not usable). All criteria received good to excellent results indicated by scores below three. Altogether, the scores for all criteria is homogeneous and result in 2.22 ± 0.51 , 2.50 ± 0.56 , and 2.41 ± 0.55 at average with the median values being 2.18, 2.55, and 2.55 for relevance, applicability, and importance of the criteria respectively. This shows the general need for all criteria.

Summarizing, we could show that the relevance criteria introduced in this paper are considered valuable to decide about data object relevance. However, this study provides only first insights with respect to the relevance and usability of the introduced criteria. A bad score in this study does not mean that the existence of a criterion is not necessary. In turn, a good score does not necessarily mean that a criterion is of use for everybody.

4 Related Work

In recent years, multiple works dealt with business process model abstraction and solved different challenges. [8] presents an approach utilizing fragments of process models as abstraction criteria consolidating them into single activities. Process modularization [1] deals with the creation of customized views by aggregating several activities into one subprocess [2] resulting in a similar approach as the fragment-based abstraction. Semantic business process model abstraction is described in [9]. The authors mainly utilize the meaning of activities, determined by natural language processing, and the data objects accessed from them to cluster the activities of the process model. Then, each cluster gets abstracted. Extending the fragment-based approach, a rule set got introduced, which allows to abstract data objects based on control flow information [5]; the abstraction of a specific activity triggers the abstraction of objects accessed by this activity.

Furthermore, there has been research to identify use cases and abstraction criteria to support business process model abstraction [10] – more specifically to support control flow abstraction. Few use cases target data objects; but these only provide information to decide activity relevance. Data object relevance is not targeted at all. The data object relevance criteria and the corresponding use cases introduced in this paper extend the existing criteria and use cases to allow control flow and data abstraction in combination as well as independently for application in business process model abstraction.

5 Conclusion

We presented eleven criteria to decide data object relevance in process models. The result can, for instance, be used in the field of business process model abstraction to allow data abstraction. The criteria are distinguished into binary ones acting as filters, continuous ones defining a threshold value and an equality condition, and hierarchical criteria, which combine a given data object to the root node of the data model accompanying the process model or to the least common denominator of all data objects used in the process model. These data object relevance criteria can be combined by defining a schema describing their connection, which may be of type sequence, conjunction, or intersection.

As identified in the evaluation, the introduced criteria cover a broad set of use cases but require further evaluation. Additionally, we will also apply the criteria to the field of business process model abstraction to allow data abstraction.

References

1. Bobrik, R., Reichert, M., Bauer, T.: View-Based Process Visualization. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 88–95. Springer, Heidelberg (2007)
2. Eshuis, R., Grefen, P.: Constructing Customized Process Views. *Data & Knowledge Engineering* 64(2), 419–438 (2008)
3. Eshuis, R., Grefen, P.: Structural Matching of BPEL Processes. In: European Conference on Web Services (ECOWS), pp. 171–180. IEEE Computer Society (2007)
4. Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. LNBIP, vol. 6. Springer (2008)
5. Meyer, A., Weske, M.: Data Support in Process Model Abstraction. In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012 Main Conference 2012. LNCS, vol. 7532, pp. 292–306. Springer, Heidelberg (2012)
6. Morasca, S.: Measuring Attributes of Concurrent Software Specifications in Petri Nets. In: Software Metrics Symposium, pp. 100–110. IEEE Computer Society (1999)
7. OMG: Unified Modeling Language (UML), Version 2.4.1 (2011)
8. Polyvyanyy, A., Smirnov, S., Weske, M.: The Triconnected Abstraction of Process Models. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 229–244. Springer, Heidelberg (2009)
9. Smirnov, S., Reijers, H.A., Weske, M.: From Fine-Grained to Abstract Process Models: A Semantic Approach. *Information Systems* 37(8), 784–797 (2012)
10. Smirnov, S., Reijers, H.A., Weske, M., Nugteren, T.: Business Process Model Abstraction: A Definition, Catalog, and Survey. *Distributed and Parallel Databases* 30(1), 63–99 (2012)
11. Weidlich, M., Mendling, J., Weske, M.: Efficient Consistency Measurement based on Behavioral Profiles of Process Models. *IEEE Trans. Softw. Eng.* 37(3), 410–429 (2011)
12. Weske, M.: Business Process Management: Concepts, Languages, Architectures, 2nd edn. Springer (2012)

Matching Business Process Models Using Positional Passage-Based Language Models

Matthias Weidlich¹, Eitam Sheetrit¹, Moisés C. Branco², and Avigdor Gal¹

¹ Technion - Israel Institute of Technology, Technion City, 32000 Haifa, Israel

{weidlich,eitams}@tx.technion.ac.il, avigal@ie.technion.ac.il

² Generative Software Development Laboratory, University of Waterloo, Canada
mcbranco@gsd.uwaterloo.ca

Abstract. Business operations are often documented by business process models. Use cases such as system validation and process harmonization require the identification of correspondences between activities, which is supported by matching techniques that cope with textual heterogeneity and differences in model granularity. In this paper, we present a matching technique that is tailored towards models featuring textual descriptions of activities. We exploit these descriptions using ideas from language modelling. Experiments with real-world process models reveal that our technique increases recall by up to factor five, largely without compromising precision, compared to existing approaches.

1 Introduction

Business process models were established for managing the lifecycle of a business process [1]. Many use cases require a comparison of process models, among them validation of a technical implementation of a business process against a business-centred specification [2], process harmonization [3], and effective search [4].

Comparison of process models involves *matching*, the construction of correspondences between activities. Such correspondences are highlighted in Fig. 1 for two models of a loan request process, defined in the Business Process Model and Notation (BPMN) [5]. The example illustrates the two major challenges of process model matching, namely textual heterogeneity and differences in model granularity. The latter leads to the definition of correspondences between sets of activities instead of single activities.

Recently, several approaches that support process model matching have been presented [2,6,7,8], inspired by techniques from schema matching [9]. These works rely on activity labels and structural or behavioural features of process models. However, they largely neglect the fact that process models are often used as documentation artefacts for which additional textual descriptions are available. Organisations provide descriptions for activities and maintain glossaries that explain the terms used in activity labels. In Fig. 1, for instance, the annotations for activity ‘*Conduct Liability Check*’ indeed support the highlighted correspondence by referring to keywords such as ‘*credit history*’ and ‘*solvency*’. Exploiting this information can improve over matching that is based only on activity labels.

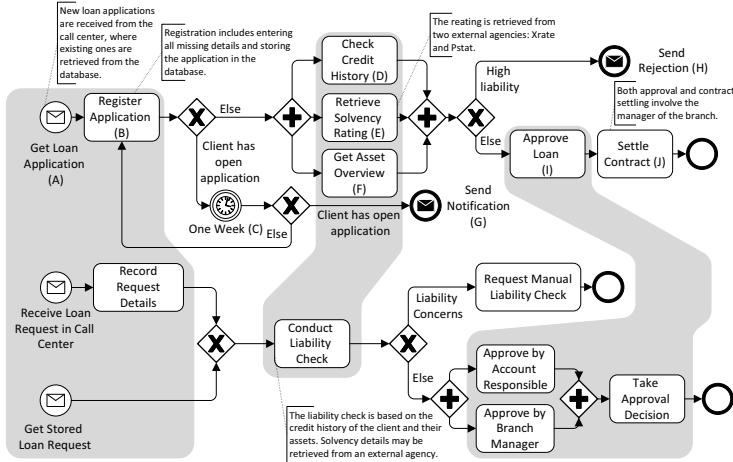


Fig. 1. An example for process model matching, correspondences are highlighted

In this paper, we present an approach to leverage textual descriptions for matching. We look at the problem from an Information Retrieval (IR) perspective. More precisely, we combine two different streams of work on probabilistic language modelling. First, we adopt passage-based modelling such that activities are passages of a document representing a process model. Second, we consider structural features of process models by positional language modelling. Our contribution is the definition of a novel language model as well as its application for process model matching. Common IR techniques are geared towards matching a single query with a collection of documents and, thus, are not applicable in our context. Hence, we also discuss how to judge the similarity of activities and derive correspondences. Our evaluation with industry process models shows that our approach can outperform existing techniques by up to a factor of five in recall, largely without compromising precision.

The rest of the paper is structured as follows. The next section provides background on language modelling. Section 3 presents our matching approach based on a positional passage-based language model. Section 4 presents an experimental evaluation. Section 5 reviews related work, before Section 6 concludes the paper.

2 Language Models for Information Retrieval

Information Retrieval (IR) extract *relevant*, often textual, information from a corpus [10] by comparing a *query* to a collection of *documents*. Recently, language models have been successfully applied in IR. In essence, they characterise a language by assigning a probability to the occurrence of a term [11]. To answer a query in IR, one first derives a language model for each document. Then, the likelihood that the query has been generated by the same language model is estimated, which yields a ranking of documents.

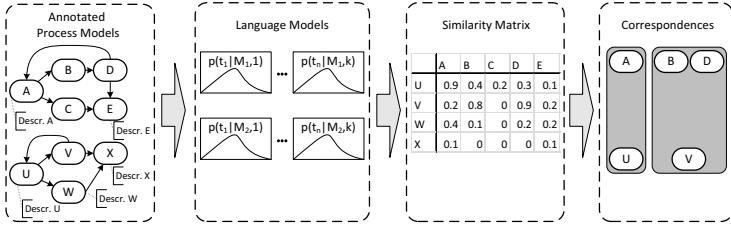


Fig. 2. Overview of the language model-based process model matching

To illustrate the basic idea, let \mathcal{T} be a set of terms and $\mathcal{B}(\mathcal{T})$ the set of all multisets over terms. Let $d \in \mathcal{B}(\mathcal{T})$ be a document with $d(t)$ as the number of occurrences of term t in d . A simple language model is a probability distribution over terms, which is based on the number of occurrences of a term in a document:

$$p(t|d) = \frac{d(t)}{\sum_{t' \in \mathcal{T}} d(t')}. \quad (1)$$

Equation 1 is independent of the importance of terms given a *corpus*, a set of documents. To countervail this effect, language models are smoothed by adding a certain probability mass to all terms that occur frequently in the corpus [12].

Our work adopts *positional* language models that define a document as a sequence of terms with a probability for a term at a document position [12]. Term proximity is integrated by propagation: term occurrences are propagated to neighbouring positions. Our approach is also inspired by *passage-based* models [13]. These models build on parts of a document identified, e.g., by section headers. As such, a passage-based model captures the probability of a term in such a passage.

3 Language Model Based Matching

The basic steps of our approach to process matching using language models are illustrated in Fig. 2. Below, we provide further details on the underlying concepts.

Positional Passage-Based Language Model. Our approach is based on a novel positional, passage-based language model. A process model is represented as a document, activities are passages of that document, and their proximity in the model is taken into account using passage positions.

For a process model P , we create a document $d = \langle T_1, \dots, T_n \rangle$ as a sequence of length $n \in \mathbb{N}$ of passages, where each passage is a set of terms $d(i) = T \subseteq \mathcal{T}$, $1 \leq i \leq n$. The set $d(i) = T$ comprises all terms that occur in the label or description of the process model activity at position i . The length of d is denoted by $|d|$. We denote by \mathcal{D} a set of processes, represented as documents.

Our model is built on a cardinality function $c : (\mathcal{T} \times \mathcal{D} \times \mathbb{N}) \rightarrow \{0, 1\}$, such that $c(t, d, i) = 1$ if $t \in T = d(i)$ (term t occurs in the i -th passage of d) and $c(t, d, i) = 0$ otherwise. To realize term propagation to close-by positions, a proximity-based

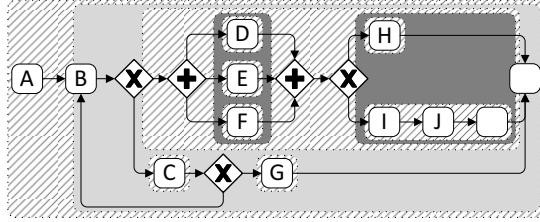


Fig. 3. Structure of the upper model from Fig. 1, normalized by an artificial end node. An example order of activities would be $A, B, E, F, D, H, I, J, C, G$

density function $k : (\mathbb{N} \times \mathbb{N}) \rightarrow [0, 1]$ is used to assign a discounting factor to pairs of positions. Then, $k(i, j)$ represents how much of the occurrence of a term at position j is propagated to position i . Lv and Zhai proposed several proximity-based kernel density functions [12]. We rely on the Gaussian Kernel $k^g(i, j) = e^{-(i-j)^2/(2\sigma^2)}$, defined with a spread parameter $\sigma \in \mathbb{R}^+$. Adapting function c with term propagation, we obtain a function $c' : (\mathcal{T} \times \mathcal{D} \times \mathbb{N}) \rightarrow [0, 1]$, such that $c'(t, d, i) = \sum_{j=1}^n c(t, d, j) \cdot k^g(i, j)$. Then, our positional, passage-based language model $p(t|d, i)$ captures the probability of term t occurring in the i -th passage of document d .

$$p(t|d, i) = \frac{c'(t, d, i)}{\sum_{t' \in \mathcal{T}} c'(t', d, i)}. \quad (2)$$

To consider importance of terms, we apply smoothing to the language model by treating each process model as a separate corpus. Then, with the corpus language model $p(t|d)$ being defined according to Equation 1, the adapted language model is defined as follows ($\mu \in \mathbb{R}$, $\mu > 0$, is a weighting factor):

$$p_\mu(t|d, i) = \frac{c'(t, d, i) + \mu \cdot p(t|d)}{\sum_{t' \in \mathcal{T}} c'(t', d, i) + \mu}. \quad (3)$$

To define how the order of passages in the document represents the order of activities in a process, we leverage the Refined Process Structure Tree (RPST) [14], a structural decomposition of a process model. The RPST is a hierarchy of non-overlapping fragments with single entry and single exit nodes. A flow arc is a *trivial* fragment; a sequence of nodes and flow arcs is a *polygon* (highlighted with striped background in Fig. 3); a fragment with multiple independent branches between the entry and exit node is a *bond* (dark solid background); other fragment structures are *rigids* (light solid background). The idea for ordering the activities is to proceed fragment-wise, starting from the root of RPST:

- If a trivial fragment has an activity as exit node, we insert the activity into the order sequence. All other trivial fragments are ignored.
- For a polygon fragments, we traverse the child fragments following the sequential order in the fragment.
- For bond fragments, we traverse the child fragments in an arbitrary order.

- For rigid fragments, we traverse child fragments as follows: starting with the entry node, we perform a depth-first traversal until we reach a node with more than one predecessor. We continue if all of these predecessors that are not reachable from the node itself (via a cycle of flows) have been visited. If not, we backtrack to the first node with multiple successors, for which not all successors have been covered and choose one of these successors randomly.

Similarity Assessment. Using the language models, we measure the similarity for document positions and, thus, activities of the process models, with the Jensen-Shannon divergence (JSD) [15]. Let $p_\mu(t|d, i)$ and $p_\mu(t|d', j)$ be the smoothed language models of two process model documents. Then, the probabilistic divergence of position i in d with position j in d' is:

$$\begin{aligned} jsd(d, d', i, j) &= \frac{1}{2} \sum_{t \in \mathcal{T}} p_\mu(t|d, i) \lg \frac{p_\mu(t|d, i)}{p^+(t)} + \frac{1}{2} \sum_{t \in \mathcal{T}} p_\mu(t|d', j) \lg \frac{p_\mu(t|d', j)}{p^+(t)} \\ \text{with } p^+(t) &= \frac{1}{2}(p_\mu(t|d, i) + p_\mu(t|d', j)) \end{aligned} \quad (4)$$

When using the binary logarithm, the JSD is bound to the unit interval $[0, 1]$, so that $sim(d, d', i, j) = 1 - jsd(d, d', i, j)$ can be used as a similarity measure.

Derivation of Correspondences. Finally, we derive correspondences from a similarity matrix over activities, which is known as second line matching [16]. Different strategies may be followed, guided by similarity values and ensuring that selected correspondences adhere to certain constraints. In our experiments, we rely on two strategies, i.e., *dominants* and *top-k*, see [16]. The former selects pairs of activities that share the maximum similarity value in their row and column in the similarity matrix. The latter selects for each activity in one model, the k activities of the other process that have the highest similarity values.

4 Evaluation

We first discuss the setup of our evaluation, before turning to the results discussion.

Setup. Our evaluation uses three real-world model collections that were also used in recent evaluations of techniques for process model matching.

BNB. This set was used by Branco et al. [2] and consists of models from the Bank of Northeast of Brazil (BNB). We selected a sample of three model pairs, all of them are in Portuguese and have few activity descriptions.

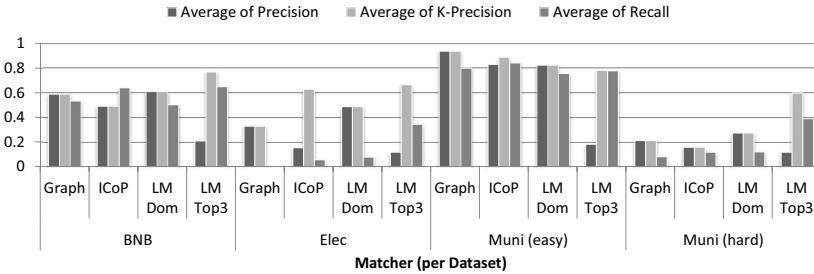
Electronics Company (Elec). This set was used by Weidlich et al. [7]. It includes three pairs of models taken from a merger in a large electronics manufacturing company, in English and with detailed descriptions of activities.

Municipalities (Muni). This collection, 17 pairs of models in Dutch with short activity descriptions, stems from municipalities in the Netherlands. Based on previous matching results [7], we separate 12 pairs representing easy matching tasks (Muni (easy)) and five pairs that are more challenging (Muni (hard)).

The used models have between 11 and 81 activities (31 on average). The gold standard was established by process analysts in the respective fields. Overall, it includes 560 matches between activity pairs.

Table 1. Results for different spreads for term propagation

Spread	BNB		Elec		Muni (easy)		Muni (hard)	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
0	0.18	0.50	0.12	0.29	0.18	0.73	0.07	0.22
1	0.21	0.65	0.14	0.31	0.19	0.78	0.12	0.39
2	0.22	0.65	0.10	0.23	0.16	0.67	0.12	0.35

**Fig. 4.** Comparison to state-of-the-art matchers for process models

To define evaluation measures, activity pairs in the gold standard are denoted by \mathcal{C}_h , those identified by a matcher are denoted by \mathcal{C}_m . Besides precision and recall, we also evaluate the extent to which our approach supports the reconciliation of the match result by an expert, which calls for high recall to reduce post-matching effort [17]. To this end, we define *k-precision* that captures in how many cases the top-k pairs proposed for an activity include a correct pair.

Precision is the fraction of selected pairs that are correct, $p = |\mathcal{C}_m \cap \mathcal{C}_h| / |\mathcal{C}_m|$. *k-Precision* extends precision to top-k lists, where a match is a top-k list where a correct pair is found: $k - p = |\{(m_1, m_2) \in \mathcal{C}_m \mid \exists (hm_1, hm_2) \in (\mathcal{C}_h \cap \mathcal{C}_m) : m_1 = hm_1 \vee m_2 = hm_2\}| / |\mathcal{C}_m|$.

Recall is the fraction of correct pairs that is selected, $r = |\mathcal{C}_m \cap \mathcal{C}_h| / |\mathcal{C}_h|$.

To compare with the state-of-the-art in process model matching, we consider two matchers. A baseline for matching based on activity labels is a graph-edit-distance based matcher (*Graph*) [6]. Second, we use a matcher of the ICoP framework [7] (*ICoP*) that uses a vector space scoring of virtual documents derived for activities and is one of the few matchers that consider activity descriptions.

Results. We first investigate the influence of the spread parameter on the match results. Table 1 (obtained with top-3 selection of correspondences) shows that a spread of 1 yields the best results in most cases. Spreads larger than 2 lower the results since a high spread blurs the characterisation of a passage.

Figure 4 compares the results obtained with existing matchers (*Graph* and *ICoP*) and two matchers based on language models, using the dominance (*LM DOM*) or top-3 (*LM Top3*) strategy for selecting correspondences. Matcher *LM DOM* yields mixed results, slightly improving over the conservative *Graph* matcher in most cases. For datasets well-addressed by existing matchers (*BNB* and *Muni (easy)*), *LM Top3* does not improve the results. However, it achieves

large improvements in recall for the challenging datasets *Elec* and *Muni (hard)*, increasing recall up to factor of 5. Although yielding low precision, the k-precision values indicate that for two-thirds of the activities in all datasets, *LM Top3* detects at least one correct corresponding pair. For instance, for dataset *Elec*, this value is comparable to the *ICoP* matcher, which is also geared towards recall. Yet, *LM Top3* identifies 5 times as many correct activity pairs for this dataset.

We conclude that language models provide a new angle for process model matching, leading to improvements for datasets for which existing tools provide poor results. These improvements come at the expense of low precision, so that the presented technique shall be applied for semi-automated matching. The high k-precision values indicate that language model-based matching is indeed suited for this setting. Reflecting on threats to validity, we note that some models had to be remodelled to achieve a common representation. Also, our models may not be representative for all scenarios of process model matching. However, our experiments covered models of three domains and in three languages, so that we expect the observations to generalize for other scenarios as well.

5 Related Work

Recent approaches to process model matching combine techniques for textual comparison of activity labels and measures for structural similarity of process model graphs. The ICoP framework [7] defines a generic architecture for this type of matchers. Following this idea, for instance, the string edit distance for activity labels has been combined with a similarity measure based on the graph-edit-distance [6], which corresponds to the *Graph* matcher in our evaluation. Other work uses the Dice Coefficient with bigrams for textual comparison of activity labels and exploits a parse tree of the process models to guide the matching [2]. Besides syntactical measures, activity labels have been compared based on semantic annotations that are derived by part of speech (POS) tagging. In [8], for instance, POS tagging of activity labels is used for deriving match hypotheses for probabilistic inference of correspondences.

Process models are often used as documentation artefacts and additional textual information is available for matching. To date, this idea was only followed by Weidlich et al. [7], applying a vector space based scoring for virtual documents derived for activities (the *ICoP* matcher in our evaluation). In this work, we took a different approach and defined a language model that allows for integrating structural details of the process model. In comparison to this previous work, our new approach leads to large improvements in recall and k-precision.

6 Conclusions

In this work, we proposed a process matching technique based on a novel combination of positional and passage-based language models. We view a process model as a document, where activity descriptions are ordered passages. We showed how these models are the basis of similarity estimation for activities and selection of

correspondences. Our evaluation shows that the presented approach is geared towards high recall, increasing it up to a factor of 5 and identifying about a third of the correct activity pairs. While average precision is low, k-precision values above 60% indicate that the correct activity pairs can be extracted by an expert with reasonable effort, thereby supporting semi-automated matching.

In future work, we intend to focus on the large differences in the results obtained for certain datasets. Here, seeking techniques for predicting the quality of match results is a promising research direction.

References

1. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: *Fundamentals of Business Process Management*. Springer (2012)
2. Castelo Branco, M., Troya, J., Czarnecki, K., Küster, J., Völzer, H.: Matching business process workflows across abstraction levels. In: France, R.B., Kazmeier, J., Breu, R., Atkinson, C. (eds.) MODELS 2012. LNCS, vol. 7590, pp. 626–641. Springer, Heidelberg (2012)
3. Weidlich, M., Mendling, J., Weske, M.: A foundational approach for managing process variability. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 267–282. Springer, Heidelberg (2011)
4. Kunze, M., Weidlich, M., Weske, M.: Behavioral similarity - a proper metric. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 166–181. Springer, Heidelberg (2011)
5. OMG: Business Process Model and Notation (BPMN) Version 2.0 (2011)
6. Dijkman, R.M., Dumas, M., García-Bañuelos, L., Käärik, R.: Aligning business process models. In: EDOC, pp. 45–53. IEEE Computer Society (2009)
7. Weidlich, M., Dijkman, R., Mendling, J.: The ICoP framework: Identification of correspondences between process models. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 483–498. Springer, Heidelberg (2010)
8. Leopold, H., Niepert, M., Weidlich, M., Mendling, J., Dijkman, R., Stuckenschmidt, H.: Probabilistic optimization of semantic process model matching. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 319–334. Springer, Heidelberg (2012)
9. Bellahsene, Z., Bonifati, A., Rahm, E. (eds.): *Schema Matching and Mapping*. Springer (2011)
10. Croft, W.B., Metzler, D., Strohman, T.: *Search Engines - Information Retrieval in Practice*. Pearson Education (2009)
11. Song, F., Croft, W.B.: A general language model for information retrieval. In: CIKM, pp. 316–321. ACM (1999)
12. Lv, Y., Zhai, C.: Positional language models for information retrieval. In: SIGIR, pp. 299–306. ACM (2009)
13. Liu, X., Croft, W.B.: Passage retrieval based on language models. In: CIKM, pp. 375–382. ACM (2002)
14. Vanhatalo, J., Völzer, H., Koehler, J.: The refined process structure tree. Data Knowl. Eng. 68(9), 793–818 (2009)
15. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory 37(1), 145–151 (1991)
16. Gal, A., Sagi, T.: Tuning the ensemble selection process of schema matchers. Inf. Syst. 35(8), 845–859 (2010)
17. Duchateau, F., Bellahsene, Z., Coletta, R.: Matching and alignment: What is the cost of user post-match effort? - (short paper). In: Meersman, R., et al. (eds.) OTM 2011, Part I. LNCS, vol. 7044, pp. 421–428. Springer, Heidelberg (2011)

Towards an Empirically Grounded Conceptual Model for Business Process Compliance

Martin Schultz

Chair for Information Systems, University of Hamburg, Hamburg, Germany
martin.schultz@wiso.uni-hamburg.de

Abstract. With the ever increasing number of legal requirements, ensuring business process compliance is a major challenge for today's organizations. Thus, compliance management gained momentum in academia and practice in recent years. Information systems (IS) researchers focus on methods providing automated support for managing diverse compliance requirements. Thereby, compliance is approached from a rather technical perspective. Little effort has been devoted to establish a comprehensive conceptualization of compliance. In particular, previous research neglected to rigorously consider stakeholders' perception based on empirical research. To close this gap, this paper presents an empirically grounded conceptual model for compliance in the context of business processes. Based on results of 17 expert interviews and an online survey, a conceptual model is constructed. The model takes into account the wide range of control means that are applied in organizations to assure compliance. Hence, the model contributes to reducing complexity and improving transparency of the compliance domain.

Keywords: Conceptual model, business process compliance, internal controls.

1 Introduction

Nowadays, organizations face the challenge of managing compliance to a steadily increasing number of rules and regulations in their everyday business operations. The rules range from voluntary norms and standards to directives imposed by active legislation. All these compliance rules significantly influence the way organizations design and execute their business processes (BP) [1]. Not surprisingly, practitioners and researchers from the process management domain (BPM) attach more importance to compliance [1]. The vast number of compliance rules and the increasing complexity of BPs in today's organizations require an (semi-)automated support for managing compliance obligations in a cost effective way [2], [3]. This especially holds true in competitive market environments where organizations strive for comprehensive standardization and automation of their BPs by extensively relying on IS. Enterprise resource planning (ERP) systems and accounting information systems (AIS) are widely used to process and store thousands and millions of business transactions each day. Hence, IS researchers have developed (semi-) automated, model-based compliance checking approaches: compliance rules are formally defined and applied to BP

models and instances [3]. These approaches consider compliance from a rather technical perspective. However, managing compliance not just involves model checking for several reasons. 1) Compliance rules can become complex, are vague and require interpretation [2]. 2) Organizations take a multitude of different measures to ensure compliance. These measures constitute a complex and interwoven system – a so called internal controls system (ICS) – that involves diverse stakeholders with varying perspectives and acting on multiple organizational levels [4]. 3) Compliance is closely linked to other organization-wide activities like corporate governance or enterprise risk management [5]. However, so far little attention has been paid to address these aspects in IS and BPM research. In particular, only few attempts have been made to conceptualize key concepts of compliance in the context of BPs [2]. This impedes a mutual understanding among stakeholders and hampers a proper formal representation as basis for designing meaningful IS support. Against this background, Sadiq (2011) identifies a “(...) well-grounded conceptual model for compliance and risk” as key issue on the research agenda for BP compliance (BPC) [2].

To close this gap, this paper outlines a conceptual model for BPC considering key concepts from an auditors' perspective as a main stakeholder of compliance. Special attention is paid to the interrelations between compliance and BP models. The design of the model is grounded on empirical research results derived from 17 expert interviews with process auditors [6] and a subsequent online survey [7] as well as a literature review of seminal research work on compliance in the IS and BPM domain.

The remainder of this paper is structured as follows. The next section outlines the related research work regarding key concepts of BPC. Section three explains the applied research method. The results of a domain analysis and related insights gained from earlier empirical research work are presented in section four. In section five the conceptual model is described. A conclusion closes the paper.

2 Related Work

Few seminal research work attempt to conceptualize compliance related concepts and their relations to BPs. Rosemann and zur Muehlen (2005) are among the first to consider the concept *risk* in the context of BP modelling [8]. They link *risk* to a generic conceptual model for BPs and provide a taxonomy of risk types. A model for BPC presented by Namiri and Stojanovic (2007, 2008) includes the concepts *risk*, *significant account*, *control objective*, *control*, and *recovery action* [9], [10]. In this model, a *control* is linked to a *process activity*, a *user*, and a *business document* which are later subsumed as so called *controlled entities*. Furthermore, a *control* mitigates a *risk* respectively supports a *control objective*. For each *control* at least one *recovery action* is defined. Different types of control are distinguished (company level control, IT control, Application control). Karagiannis et al. (2007) present a solution for Sarbanes-Oxley Act reporting requirements and consider *risk*, *control*, and *account* as domain specific concepts. These are linked to *BP elements* (including IS, BP activity, organizational unit). *Control objective*, *control*, and *risk* are also set in relation by Lu et al. (2008) [11]. Similarly, Strecker et al. (2011) stress *control objective* and *control*

means as main concepts in a conceptual model for an internal control system [4]. *Control objectives* are linked to *risk*, *goal*, and *codification*. *Control means* support a *control objective* and are realised by a BP, organizational unit, and/ or IS. The concepts are introduced as an extension to an existing enterprise modelling approach. Sadiq et al. (2007, 2009, 2010) use the concepts *control objective*, *internal control*, as well as *risk* and relate these to *process*, *task*, and *property* for an ontological alignment between compliance and the BP domain [12], [13], [14]. Schumm et al. (2010) present a rather abstract conceptual model focusing on *compliance requirements* that stem from a *compliance source*, relate to a *compliance risk* and can be assessed by a *compliance request* [15]. A *compliance requirement* can be addressed by a *control* that is formally expressible as a *compliance rule* and refers to an abstract *compliance target* (BP, BP element). A similar model is presented by Turetken et al. (2011) [16]. Table 1 summarizes the key concepts identified in earlier research work.

Table 1. Overview of domain specific concepts for BPC in related work

Domain Concept	Mentioned by Authors	Muehlen (2005)	Namiri (2007)	Karagiamnis (2007)	Sadiq (07/09/10)	Lu (2008)	Schumm 2010	Strecker (2011)	Turetken (2011)
Control Objective/ Compliance Requirement	●			●	●	●	●	●	●
Risk	●	●	●	●	●	●	●	●	●
Compliance Source/ Codification						●	●	●	●
Compliance Target/ Controlled Entities	●					●		●	●
Compliance Rule						●		●	●
Control (Means)	●	●	●	●	●	●	●	●	●
Recovery Action	●					●	●	●	●
Compliance/ Risk Assessment/ Request	●	●				●	●	●	●
Compliance Fragment						●			
Compliance Concern								●	
Business Process	●	●	●	●	●	●	●	●	●
BP element (activity, document, user, IS)	●	●		●	●			●	
Goal	●						●		
Financial Account	●	●	●						

3 Research Method

The research presented in this paper follows the design science approach [17]. The designed artefact is a conceptual model comprising relevant concepts of BPC and their relations. In earlier work we applied a multi-method research approach by combining a qualitative (expert interviews) and a quantitative (online survey) research method to rigorously identify key concepts of the domain from a stakeholder point of view. Regarding the construction of the model, we explicitly elaborate on specific design decisions to ensure transparency. The relevance of the artefact stems from the fact that methods for managing compliance in a cost effective way is an urgent need of many organizations [2], [3]. However, a well-grounded conceptual model as basis for developing appropriate methods and IS support is still missing [2].

There is a consensus in literature that evaluating a designed artefact is an essential step in design science research. As an initial evaluation step the conceptual model is used to design another IS artefact for auditors. Due to page limitations the evaluation is not included in this paper. The evaluation will be supplemented by future research.

4 Domain Analysis

4.1 Terminological Analysis

Designing a conceptual model presupposes the reconstruction of key terms and concepts of the targeted domain [4]. In a broad perspective, compliance describes a state of an organization regarding the conformance to a set of regulations and rules or represents a process to ensure this conformance. The norms originate from a wide range of sources ranging from laws and regulatory requirements to internal guidelines [16], [18]. Compliance is closely linked with the ICS an organization has to maintain. Internal control is broadly defined as a process designed to provide reasonable assurance regarding the achievement of objectives in three categories: 1) effectiveness and efficiency of operations, 2) reliability of financial reporting, and 3) compliance with applicable laws and regulations [19]. It is a system of integrated elements like people, organizational structures, processes, and procedures and therefore covers procedural as well as structural aspects [4]. Key concepts are control objectives and control means. A control objective describes a desired state of an organization or a process and is associated with a recommended course of action that should be taken (control means) to ensure that a control objective is achieved. Control means may involve policies, procedures, practices (e.g. reviews, checks) as well as organizational structures (e.g. authorizations, roles, organizational units) [4], [19], [20].

Audit standards distinguish between process-integrated and process-independent control means (e.g. internal audit function of an organization) [20]. Process-integrated means are further specified as organizational means and control means. Organizational means are preventive security measures that are integrated in the organizational and operational structure of an organization e.g. restricted access, segregation of duties (SoD), approval levels. Control means in this context are measures that are directly integrated in the sequence of operations and constitute e.g. check for completeness or validity [20]. In this procedural sense, a control means represents a target/actual performance comparison enacted as a preventive or detective activity in a process [21]. The terminological analysis reveals that the term control means is subject to terminological ambiguity as it denotes not only procedural but also structural aspects [4]. This is a notable differentiation that is covered in audit standards but so far not comprehensively considered in IS-related literature concerning BPC.

Another core concept strongly related to compliance is risk. Risk can be broadly described as a threat to the achievement of entity's goals/ objectives. To measure risks probability of occurrence and impact of a threat are usually used. A risk has reference objects for which organizational goals are set (e.g. BP) [8], [19], [22]. The relation between risk and compliance is twofold. On the one hand, compliance requirements often directly refer to specific risks. At the same time, compliance requirements

introduce a new risk, namely the risk of non-compliance or compliance risk [18]. On the other hand, control objectives are defined and control means are implemented to mitigate risks by reducing their probability of occurrence and/ or their impact.

Compliance is also closely related to the audit domain. Auditing an organization's compliance is often a legal requirement, i.e. annual audits of the financial statements. As audit standards enforce an in-depth analysis of the organization's operations, BPs constitute a central audit subject. Auditors focus on the ICS of an organization as well-controlled BPs contribute to a compliant state of an organization [20].

4.2 Results of Empirical Domain Analysis

For integrating rigorous empirical evidence in the construction process of a conceptual model for compliance, we conducted 17 semi-structured expert interviews and a subsequent online survey (370 respondents) among internal and external auditors [6], [7]. By doing so, we applied a multi-method research approach to determine their understanding on key concepts in the context of process audits. Methodological details on the conducted empirical research are outlined in the respective papers [6], [7].

The results demonstrate a consistent understanding among the auditors regarding the concepts that need to be considered in the context of BPC. 12 concepts are derived: *audit/ control objectives*, *control means*, *risk*, *audit results*, *standards & regulations*, *financial statements*, *materiality*, *business objectives/ goals*, *process flow*, *information systems*, *organization*, and *data*. These concepts correspond (with partially different terms) to the results of the terminological analysis (section 4.1) and the concepts discussed in related research work (section 2) except the concepts *audit results* and *materiality*. Regarding the concept *control means* the analysis of the empirical data reveals, that auditors consider control means as a special activity in a process that need to be regularly conducted to ensure compliance. The results regarding relations among the concepts are more differentiated. There are only a few relations clearly classified as relevant by the majority of experts and respondents.

5 Conceptual Model for Business Process Compliance

As outlined in section 2, there is consensus on a certain set of concepts that need to be considered when dealing with BPC. Fig. 1 depicts the conceptual model with these concepts identified for the BP and the compliance domain as a class diagram. The model clearly separates these two domains to underline their various relations. This separation facilitates traceability between compliance concepts and related BP concepts [16]. We include the concepts 1) identified as relevant in our empirical work and 2) supported by the majority of authors of related work. Accounting specific concepts (financial accounts/ statements, materiality) are not considered to abstract from specific compliance sources and keep the model generally applicable. The relations are based on the terminological analysis of the domain. In the following the concepts and their relations are briefly described. Specific design decisions are outlined in more detail. From a compliance perspective for BPs a rather generic model can be

assumed consisting of the concepts process (control flow), process activity, and other process elements (organizational resource, data object and IS) [16]. Additionally, the process supports particular organizational goals. The depicted concepts and relations comply with existing meta models for BPs [8]. Using such a generic model ensures that the conceptual model is not restricted to particular BP modelling techniques [16].

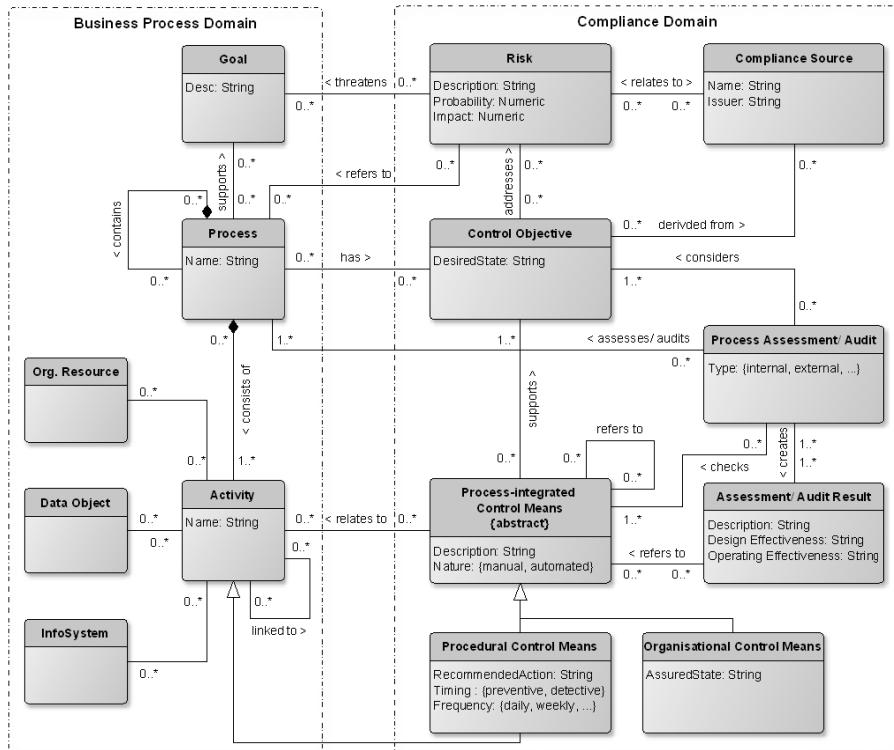


Fig. 1. Conceptual Model for Business Process Compliance

A *Control Objective* describes a desired state of a *Process* and is derived from a *Compliance Source* (e.g. standards, regulations, laws) and/ or addresses a *Risk*. The concepts compliance source and risk are linked to each other as a compliance source can relate to specific risks and might introduce new risks (compliance risks [18]). A risk is linked to a *Goal* as it threatens its achievement. There are one or more *Control Means* supporting a particular control objective. To account for the multiplicity of control means that can be used to achieve a control objective we decide to include an abstract class *Process-integrated Control Means*. The recursive association allows representing control means as interconnected system of control means. In accordance with audit standards, two types of control means inherit from this abstract class [20]. Firstly, *Organizational Control Means* represent all organizational means that can be implemented to ensure compliance, e.g. restricted access, SoD, approval levels. These means refer to requirements that can be easily expressed as a formal rule a process

activity has to comply with (e.g. “access to activity X is restricted to appropriate personnel”). Secondly, *Procedural Control Means* refer to all measures that involve a recommended course of action within the process to ensure compliance (e.g. reconciling sub and general ledger on a weekly basis). These control means significantly differ from the organizational control means as they constitute activities in a particular process. Therefore, this class also inherits from the class *Activity*. Doing so, these control means can be part of a process as a specific type of activity. This conceptual design provides an important link between compliance and BP, reflects the results of the terminological analysis and our empirical results, reduces terminological ambiguity and improves transparency when referring to different types of control means with distinct properties [4]. In a *Process Assessment* of a process related control means are checked against a set of control objectives. An assessment provides an *Audit Result*.

6 Conclusion

Due to an ever increasing number of regulatory requirements today’s organizations have to comply with in their daily business, compliance management gained momentum in practice and in academia in recent years. So far, IS researchers consider BPC as rather technical matter and focus on methods to provide (semi) automated support for managing compliance requirements. Only few attempts have been made to establish a comprehensive conceptualization for BPC. Especially, little effort has been devoted to empirical research to rigorously consider stakeholders’ perception of this complex domain. To close this gap, this paper presented an empirically grounded conceptual model for BPC. Based on the results of 17 expert interviews and an online survey among internal and external auditors as well as a literature based domain analysis, a conceptual model was constructed. The model takes into account the various types of control means that can be applied to fulfil a certain compliance requirement.

There are several opportunities for further research work. The research presented here is limited to auditors as one stakeholder group for compliance. By considering further stakeholders, valuable new insights could be derived providing a fuller picture of the domain. Similarly, a multi-perspective approach for evaluating the conceptual model contributes to the body of knowledge. This remains on our research agenda.

References

1. Liu, Y., Muller, S., Xu, K.: A static compliance-checking framework for business process models. *IBM Syst. J.* 46, 335–361 (2007)
2. Sadiq, S.: A Roadmap for Research in Business Process Compliance. In: Abramowicz, W., Maciaszek, L., Węcel, K. (eds.) *BIS Workshops 2011 and BIS 2011. LNBP*, vol. 97, pp. 1–4. Springer, Heidelberg (2011)
3. Becker, J., Delfmann, P., Eggert, M., Schwittay, S.: Generalizability and Applicability of Model-Based Business Process Compliance-Checking Approaches – A State-of-the-Art Analysis and Research Roadmap. *Bur - Bus. Res.* 5, 221–247 (2012)

4. Strecker, S., Heise, D., Frank, U.: Prolegomena of a modelling method in support of audit risk assessment - Outline of a domain-specific modelling language for internal controls and internal control systems. *Enterp. Model. Inf. Syst. Arch.* 6, 5–24 (2011)
5. Racz, N., Weippl, E., Seufert, A.: A Frame of Reference for Research of Integrated Governance, Risk and Compliance. In: De Decker, B., Schaumüller-Bichl, I. (eds.) CMS 2010. LNCS, vol. 6109, pp. 106–117. Springer, Heidelberg (2010)
6. Schultz, M., Mueller-Wickop, N., Nuetgens, M.: Key Information Requirements for Process Audits – an Expert Perspective. In: Proceedings of the 5th EMISA, Vienna, Austria, pp. 137–150 (2012)
7. Mueller-Wickop, N., Schultz, M., Peris, M.: Towards Key Concepts for Process Audits – A Multi-Method Research Approach. In: Proceedings of the 10th ICESAL, Utrecht, The Netherlands, pp. 70–92 (2013)
8. Rosemann, M., Muehlen, M.Z.: Integrating Risks in Business Process Models. In: Acis 2005 Proc. (2005)
9. Namiri, K., Stojanovic, N.: Pattern-Based Design and Validation of Business Process Compliance. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 59–76. Springer, Heidelberg (2007)
10. Namiri, K., Stojanovic, N.: Towards A Formal Framework for Business Process Compliance. *Multikonferenz Wirtsch.* 2008, 259 (2008)
11. Lu, R., Sadiq, S., Governatori, G.: Compliance Aware Business Process Design. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 120–131. Springer, Heidelberg (2008)
12. Sadiq, W., Governatori, G., Namiri, K.: Modeling Control Objectives for Business Process Compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 149–164. Springer, Heidelberg (2007)
13. Sadiq, S., Governatori, G.: A methodological framework for aligning business processes and regulatory compliance. *Handb. Bus. Process Manag.* 2, 159–176 (2009)
14. Sadiq, S., Governatori, G.: Managing Regulatory Compliance in Business Processes. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management, vol. 2, pp. 159–175. Springer, Heidelberg (2010)
15. Schumm, D., Turetken, O., Kokash, N., Elgammal, A., Leymann, F., van den Heuvel, W.-J.: Business Process Compliance through Reusable Units of Compliant Processes. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 325–337. Springer, Heidelberg (2010)
16. Turetken, O., Elgammal, A.: Enforcing Compliance on Business Processes through the Use of Patterns. In: Proceedings of the ECIS 2011, Helsinki (2011)
17. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *Mis. Q* 28, 75–105 (2004)
18. Klotz, M., Dorn, D.W.: IT-Compliance–Begriff, Umfang und relevante Regelwerke. *Hmd-praxis Wirtsch* 263, 5–14 (2008)
19. COSO: Internal Control - Integrated Framework (1992), <http://www.coso.org>
20. IAASB: ISA 315 - Identifying and Assessing the risks of Material Misstatement through Understanding the Entity and its Environment. International Auditing and Assurance Standards Board (2009)
21. Sackmann, S., Hofmann, M., Kühnel, S.: Return on Controls Invest. *Hmd - Prax. Wirtsch.* 289, 31–40 (2013)
22. Strecker, S., Heise, D., Frank, U.: RiskM: A multi-perspective modeling method for IT risk assessment. *Inf. Syst. Front.* 13, 595–611 (2011)

Improving Business Process Intelligence with Object State Transition Events

Nico Herzberg, Andreas Meyer, Oleh Khovalko, and Mathias Weske

Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert-Str. 2–3, D-14482 Potsdam, Germany
{nico.herzberg, andreas.meyer,
oleh.khovalko, mathias.weske}@hpi.uni-potsdam.de

Abstract. During the execution of business processes several events happen that are recorded in the company's information system. These events deliver insights into process executions so that process monitoring and analysis can be performed resulting, for instance, in prediction of upcoming process steps or the analysis of the runtime of single steps. While event capturing is trivial when a process engine with integrated logging capabilities is used, manual process execution environments do not provide automatic logging of events, so that typically external devices, like bar code scanners, have to be used. As experience shows, these manual steps are error-prone and induce additional work. Therefore, we use object state transitions as additional monitoring information, so-called object state transition events. Based on these object state transition events, we reason about the enablement and termination of activities and provide the basis for process analysis in terms of a large event log.

Keywords: Business Process Management, Events, Data, Process Monitoring, BPMN.

1 Introduction

Nowadays, companies face a very competitive market environment. Therefore, they strive to run their value generating operations in a process-oriented way to stay competitive and to be able to react on market changes quickly. These business processes are managed by using techniques and methodologies of business process management (BPM). BPM deals with the organization, documentation, analysis, optimization, and execution of business processes [23]. One important aspect of BPM is business process intelligence (BPI) that comprises process analysis, monitoring, and mining [2, 5]. BPI requires information about process behavior and events that occur during process execution. Process monitoring is used to predict upcoming events and process steps by observing the process execution and deriving the corresponding process behavior. Monitoring is usually applied on process models being enacted by a process engine - an information system that controls the process execution - because it generally provides logging capabilities such that the current process progress is easily recognizable from the observed events.

The observation of such an event determines the current position in the course of process execution. In contrast, manually executed processes, as, for instance, usual in

health care, lack these capabilities such that most events cannot be observed or stored, i.e., event logs are incomplete. However, recognition and prediction of process progress as well as a holistic view on the process execution shall be enabled in these domains also. Introducing additional external devices and similar logging equipment is not applicable in manual processes – especially in time critical ones as in health care – due to extra amount of work and complexity. Therefore, event monitoring points [6] are defined to specify where in the process events are expected. Event monitoring points can be considered as milestones indicating the achievement of important business value.

In this paper, we apply the concept of event monitoring points to data objects to create more expressive event logs without adding additional logging mechanism or devices such as scanners, RFID tags, or sensors. We further elaborate on the ideas about the concept of *object state transition events* (in short: *transition events*), which have been initially sketched in [7]. Thereby, we reason about activity termination (enablement) by means of events indicating the data objects to be written (read) by that activity transitioned into the respective object state and vice versa. Additionally, this technique also increases the number of events observed in enacted process models such that analysis is based on a much higher number of inputs resulting in an increased reliability of the results.

The remainder of the paper is structured as follows. In Section 2, we introduce the foundations combined with the scope of our approach followed by an approach overview and detailed descriptions of its application. Section 3 and Section 4 describe the application of the introduced approach to a use case during *design time* (model view) and during *run time* (instance view). Finally, we discuss related work in Section 5 and conclude the paper in Section 6.

2 Approach

In this section, we will introduce the fundamental notions for and the scope of our approach, which we will introduce high-level in Section 2.2. Afterwards, we will present details about the main components of our approach, starting with the concept of object state transitions, naming the challenges and showing solutions for that, in Section 2.3. The notion of binding is presented in Section 2.4 followed by the description of the process modeling aspects of the approach in Section 2.5.

2.1 Foundations and Scope

Within our approach, we connect process models with object life cycles, which describe the actions and manipulations performed upon data objects. Thereby, a data object is any piece of information being processed, manipulated, or worked with during process execution. Each data object can exist in several states, which represent specific business situations of interest from the data object's point of view. The dependencies of these object states are defined by means of object life cycles.

Definition 1 (Object Life Cycle).

An *object life cycle* is a tuple $L = (S, Z, i, T, \gamma, \Sigma, \eta)$, where S is a finite non-empty set of *object states*, $Z \subseteq S$ is a finite set *final object states*, $i \in S \setminus Z$ is the *initial object*

state, $T \subseteq S \times S$ is a finite set of *object state transitions*, function $\gamma : S \times S \rightarrow T \cup \emptyset$ maps each pair of object states to the transition connecting them, if such transition exists, or the empty set otherwise, Σ is a finite set of *labels* (S, T, Σ are disjoint), and function $\eta : T \rightarrow \Sigma$ assigns to each object state transition the corresponding label representing the action initiating that transition. \diamond

We represent an object life cycle as a state machine, where the object states are represented by elliptic nodes and the transitions are represented by edges connecting two states. Each edge is labeled with the action leading to the state change.

Those state transitions might be represented through event objects during run time. We refer to an event object as real-world happening occurring in a particular *point in time* at a certain *place* in a certain *context* that is represented in the information system landscape [8]. Each event object can be correlated to any positive number of nodes or object state transitions and is stored in an IT system such that they are accessible as *(semi-) structured data*. In the scope of this paper, we focus on event objects, which can be correlated to object state transitions. Thereby, we assume that all expected events are recognized and stored, i.e., we exclude the discussion of missing events from this paper.

By using data objects and their states in process models, these data objects are utilized during process execution. Thus, the event objects relating to the data objects' life cycles are relevant for the corresponding process execution. We formally define the process model and set the scope about the process models we support with our approach.

Definition 2 (Process Model).

A *process model* is a tuple $M = (N, D, R, C, F, type)$, where $N \subseteq (A \cup G \cup E)$ is a finite non-empty set of *nodes*, which comprises sets of activities A , gateways G , and events E . D is a finite non-empty set of *data objects*, R is a finite non-empty set of *object states*, $C \subseteq (A \cup G \cup E) \times (A \cup G \cup E)$ is the *control flow* relation, $F \subseteq (A \times (D \times R)) \cup ((D \times R) \times A)$ is the *data flow* relation, and $type : G \rightarrow \{xor, and\}$ assigns to each gateway a type. \diamond

Let S_d be the set of object states defined in the object life cycle for a data object $d \in D$, then the super set of object states $\mathcal{S}_M = \sum_{d \in D} S_d$ for process model M denotes the set of object states specified in the object life cycles of all data objects being associated to an activity of M . The set of object states R used in M is a subset to the super set of object states, i.e., $R \subseteq \mathcal{S}_M$. Further, we require each process model M to fulfill basic structural requirements. First, the process model may be arbitrarily structured but it needs to be structural sound [1], i.e., M contains exactly one start and one end event and every node of M is on a path from the start to the end event. Further, each activity and event has at most one incoming and one outgoing control flow edge, while each gateway has at least three incident control flow edges with at least one incoming and at least one outgoing control flow edge. The events added to a process model only represent a small fraction of the events actually occurring during process execution such that most events are not comprised in the model but occur during process execution.

The control flow semantics of the process model follows Petri net semantics [17]. The data flow semantics inherits the concept of data input and output sets from BPMN [21]. The data input set contains a positive number of sets, each specifying the data objects in specific object states required to enable the activity. If one of the sets is

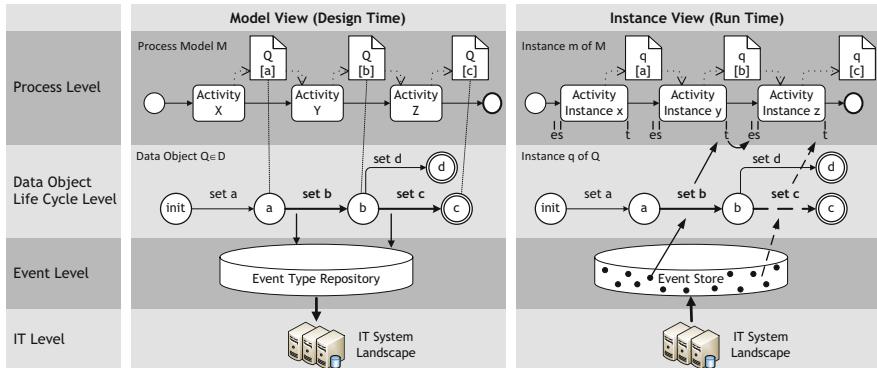


Fig. 1. Overview of the approach at design time and run time

completely satisfied by the existing data objects, activity enablement takes place. Otherwise, process execution is paused until satisfaction is observed. The check for data object existence for a certain activity is only done, if the control flow approaches that activity. Following, we assume activity enablement requires control flow and data flow enablement at the same time. Analogously, the data output set specifies different options of which data objects in which object states exist after activity termination. The corresponding set is chosen during the execution of the respecting activity. Based on these data flow semantics, we assume that the observation of a data object write (the data object exists in the specific object state) indicates activity termination and vice versa.

Finally, we also assume that the process model only utilizes object state transitions comprised in the object life cycle, i.e., the combination of process model and object life cycles fulfill the notion of weak conformance [13]. The object life cycle may contain object states and transitions not utilized in the process model.

2.2 Methodology

The presented approach enables process execution analysis by utilizing the information about data objects and their life cycles. Thus, the required data objects and their life cycles need to be modeled first. In Fig. 1—Data Object Life Cycle Level, we provide an example with one data object. Assume, data object $Q \in D$ may appear in five different states (*init*, *a*, *b*, *c*, and *d*), which are reached by the state transitions *set a*, *set b*, *set c*, and *set d*. For the given life cycle, the observable state transitions need to be selected; a transition to a particular object state is observable, if it can be monitored by a specific event named *object state transition event* (see Definition 3). In Fig. 1, we denote observable object state transitions with bold (*set b*) and dashed (*set c*) lines.

Definition 3 (Object State Transition Event)

An object state transition event $\mathcal{E} = (\text{type}, \text{id}, \text{timestamp}, q, s)$ consists of an *object state transition event type* referencing the corresponding data object $Q \in D$, a unique *identifier*, a *timestamp* indicating the point in time when the object transition occurred, the instance q of the data object Q , and the object state s of object Q reached when the event was triggered. \diamond

An object state transition event \mathcal{E} is an instance of an *object state transition event type* (see Definition 4) that describes the creation logic for the underlying transition events based on the corresponding data object. The information about the object state in the transition event is necessary to enable the correlation of the object state transition to the corresponding activity in the business process. The transition event holds a snapshot of the data object at the point in time a particular object state is reached. This is required to enable process analysis with respect to certain process data as, for instance, object attributes. Assume a stakeholder is interested in the costs of activities executed in the course of process model M , see Fig. 1, and data object Q contains an attribute in which execution costs are summed up. For a specific process model instance m of M and therefore a specific data object instance q of Q , the required process analysis would be about the costs recorded at every step in the data object instance q . Another analysis targeting execution times allows identifying bottlenecks in the process model and both mentioned analyses may visualize room for process improvements.

The object state transition event aligns to structured events as defined by Herzberg et al. [8] with the event content from their definition being represented by the snapshot of the data object q and data state s reached through the respecting transition. Referring to this work, we define an *object state transition event type* \mathcal{ET} for each data object that is managed in an event type repository, Fig. 1—Event Level.

Definition 4 (Object State Transition Event Type)

An object state transition event type is a tuple $\mathcal{ET} = (Q, B)$, where $Q \in D$ is the corresponding data object and B is a set of bindings that are required to access the information about the state transitions of the particular data object Q . ◇

The event type refers to the data object affected by a state transition represented by an object state transition event. Additionally, the object state transition event type holds information about the set of all *bindings* available for the particular data object. A binding links an object state transition of a specific object life cycle with an information system allowing the identification of the respecting state change information, because information about the transition of a data object to a new object state is represented in the information system landscape as shown in Fig. 1—IT Level. Such state change information may be, for instance, the insertion or update of an entry with a specific identifier in a database. An object state transition is observable if and only if there exists a binding for that transition.

During process modeling, data objects may be associated with several process models. In our example, see Fig. 1—Process Level, data object Q is associated with process model M and is read and written in different states (a , b , and c) by particular activities, which are in sequence X , Y , Z surrounded by a start and an end event. An activity reads a data object in a certain state, creates a data object in a certain state, or transfers a data object from one state into another one. For instance, activity Z transfers Q from state b to state c only although the object life cycle would also allow a transition to state d . Altogether, the given process model conforms to the given object life cycle [13].

During the process execution (run time) the object state transition events are created according their specification described by the object state transition event type. The *event store* (Event Level in the Instance View in Fig. 1) comprises all transition events generated and persists them for further usage. Information about process execution can

be gathered from the assignment of data objects to activities and the transition events. We assume that an activity is enabled if the activity could be executed with respect to the control flow specification and if all required input data objects are available in the corresponding object states as specified in the data object input set. Analogously, we assume that an activity is terminated once the data objects are present in the respective object states as specified in the data object output set. A data object input respectively output set specifies for each activity the input respectively output data objects including their object states and their required combination.

As aforementioned, process execution can be monitored for those activities that consume respectively provide data objects in certain object states, where the corresponding object state transitions can be observed by events. In Fig. 1, the object state transition *set b* is observable because of an existing binding and event type. If a corresponding event appears in the event store, we can reason about activity termination for the corresponding process instance. The object state transition *set a* cannot be observed such that it is not possible to derive information about the termination of activity *X* through data object information. Knowing about the existence of output data objects and the termination of the corresponding activity allows to derive the enablement of the directly succeeding activity from that information, if the succeeding activity requires a subset of the output data objects from the preceding activity. In Fig. 1, observing a transition event correlating to the transition *set b*, we consider activity *Y* of the respecting process instance to be terminated and activity *Z* gets enabled, because it uses exactly the output of *Y* as input. In the figure, we denote this insight by the bent arrow from *t* to *e* between both activities in the instance view. Furthermore, an event correlating to transition *set c* is not yet being observed although it is expected to happen at some point in time. This, we indicate with a dashed bold line between object states *b* and *c* in the object life cycle.

2.3 Object State Transitions

Object state transitions highly influence activity enablement and termination that are closely coupled with data object input and output sets respectively, which are specified for each activity. Both sets are comprised by further sets with each set containing a positive number of data objects in specific object states. Fig. 2 represents a process model fragment and an object life cycle comprising the state manipulations done by the activities. Activity *A* and *B* are executed in interleaving order; the actual execution order is not enforced by data dependencies due to the extensive data object input and output alternatives and therefore relies on control flow only. In Fig. 2a, the input set $\{\{(Q, b)\}, \{(Q, x)\}\}$ of activity *A* comprises two sets each containing one entry stating that data object *Q* is required for enablement either in state *b* or in state *x*. In case the input set comprises multiple sets, they will

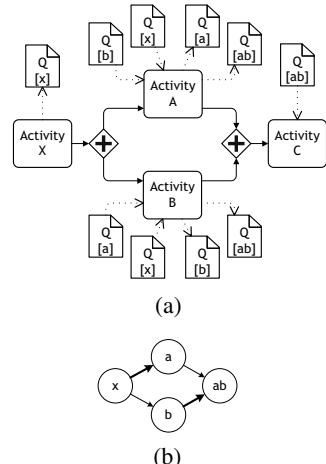


Fig. 2. Process fragment (a) and corresponding object life cycle (b)

be checked one by one for satisfaction and the first one satisfied by currently existing data objects is utilized to enable the corresponding activity.

As already mentioned above, only state transitions with a respecting binding are observable. Referring to Fig. 2b, only the transitions corresponding to activity A in Fig. 2a are observable. Following, execution of activity B cannot be captured by means of object state transition analysis such that we cannot easily reason about enablement of activity C. In fact, between termination of activity X and termination of activity C, we cannot identify the exact current state of the process, i.e., which activity is currently enabled. But, combining the approach presented in this paper with control flow monitoring approaches, the execution of B might be detected through a positioned event monitoring point, preferably to monitor termination of B. Then, we can easily reason about the execution of both interleaving activities and following about enablement of C. Knowing about the existence of output data objects (here: from B) and based on that control flow information, we can derive that a state transition corresponding to activity B must have been occurred. If the event monitoring point is related to another happening of B, e.g., enablement or begin, above mentioned reasoning only leads to success if B is executed before A.

Assuming the life cycle from Fig. 2b with state ab being independent from states a and b and an activity Z having state x as input and state ab as output, two state transitions are comprised by that activity. If both transitions to state ab can be observed, we can reason about termination of Z. Otherwise, control flow information needs to be considered (see above). In case only one transition to ab is observable but a previous transition on the other path is also observable (as notated in Fig. 2b), we can derive at run time whether reasoning about termination of Z is possible from object information. If the transition to state x is not observable and therefore enablement of Z cannot be implied, observing of state a allows to reason about the start of Z.

2.4 Binding

Organizations, which follow a model driven execution of business processes, collect the information about the execution of their processes in information systems. In our approach, we assume that information about events is extracted from existing legacy data stores. All data processed during the process execution is stored in the IT systems including process context data, processed data objects, and occurred events.

In this section, we discuss how the object state transitions represented within a Data Object Life Cycle Level can be mapped to the corresponding data state representations at the IT Level (see Fig. 1). For this purpose, we define the *binding* of a specific object state transition of a data object to the corresponding object state representation as follows:

Definition 5 (Binding)

A *binding* is a mapping function $bind : D \times T \rightarrow I$, where D is a finite non-empty set of *data objects*, T is a finite set of *object state transitions* modeled within a data object life cycle, and I is the set of *implementations*, i.e., rules and methods, specifying how to extract event information from different data sources within the information system landscape. ◇

The defined binding-function can be applied for various data sources within an IT systems landscape, e.g., for querying databases, processing Web service invocations, ana-

lyzing structured datasets, stream processing filters, or reading a log entry. Assuming, for instance, that event information is stored within a relational database, the techniques of schema summarization [24] combined with schema matching techniques [22] and matching approaches based on linguistic comparison [22] can be applied to map the object state transitions to the corresponding data representation and to realize the binding – in this particular example – to columns and rows of a relational database. Thus, the *bind* function is used at *design time* to identify the mappings, which are then stored within the binding repository.

In the second step, the object state transition events are generated based on the IT Level representation at *run time*, using the mapping information, which is defined by the binding function. Each discovered event will then be stored within an event store using the structure defined in Definition 3.

2.5 Enable Object State Transition Events by Process and Data Modeling

For our approach, we allow independent modeling of data objects and their life cycles and of business processes. However, data objects are required to be modeled in the process model as well – to enable more detailed views on the business processes [14] and to allow the application of the approach introduced in this section. The independent modeling allows reusing data objects and their object life cycles such that they may be modeled once with all applicable manipulations for each object and then reused by coupling them to multiple process models using a subset of the specified manipulations, i.e., state transition. Thereby, a process model must only use a set of transitions of the corresponding object life cycle for single activities, i.e., each process model has to satisfy the notion of weak conformance with respect to all data objects used in that model [13].

Summarizing, the presented approach to improve business process intelligence consists of several steps: (i) the definition of an object life cycle for each data object using a state machine, (ii) the specification of the observable object state transitions in the object life cycle via the binding function, which maps such transition into an information system and (iii) the model of the business process and the assignment of data objects to the corresponding activities. Utilizing this setup, processes can be partly monitored by analyzing the data object states of activities' input respectively output sets to predict the enablement respectively termination of those events. In the next section, we will apply the approach to an order process.

3 Model View Application by Use Case

We apply the presented approach to a product order process. A product is requested from the customer by an order. This request could be a normal one or an urgent one that requires express shipping. The orders are fulfilled with products in stock; however, sometimes the products need to be produced by machines first. Thus, we find four different data objects in this example case: (a) the order, (b) the product, (c) machines in the production line, and (d) the type of shipment. These data objects are described by object life cycles, see Fig. 3.

Based on these object life cycles the state transitions are selected that can be observed in the information systems and the bindings are defined that allow the access to this data.

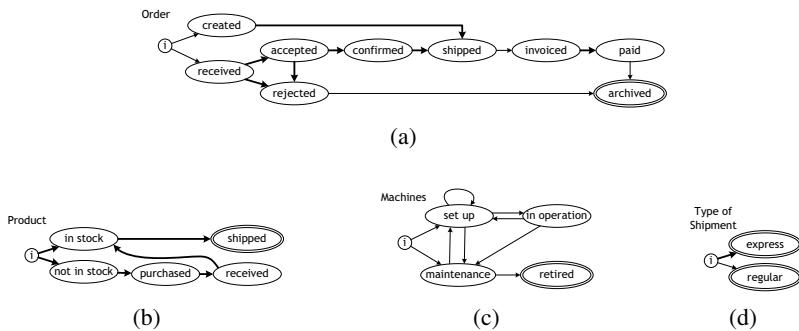


Fig. 3. Life cycles of data objects (a) Order, (b) Product, (c) Machines, and (d) Type of Shipment used in the process model in Fig. 4. The transition labels are omitted due to space requirements.

Table 1. Example of representation of the *Product* object instances in a relational database

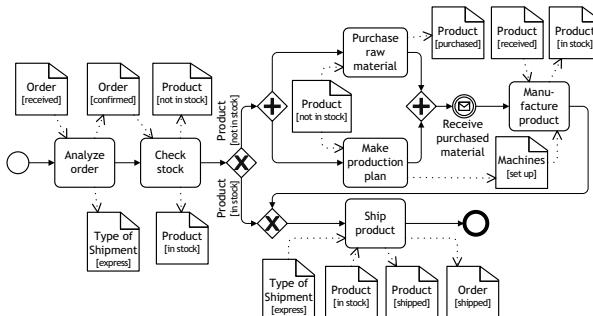
product_id	...	in_stock	not_in_stock	purchased	received	shipped
856	...	2012-12-12 16:02	2012-12-11 10:40	2012-12-11 10:51	2012-12-12 15:13	null
857	...	2012-12-14 08:34	null	null	null	2012-12-19 13:01
858	...	2012-12-19 11:55	null	null	null	null

For orders it is possible to observe the data state transitions to states *accepted*, *rejected*, *confirmed*, *shipped* and *paid*. In case of the product, every state transition can be tracked in the information systems, whereas for the machines none of the state transitions are tracked. For the data object *type of shipment* the state transition to the state *express* is observable.

Consider the representation of the data object *Product* depicted exemplarily in Table 1. The presented table shows excerpt-wise a table of a relational database, which stores the states of the object *Product*. We omit some columns for space reasons. Each row of the table represents an instance of the *Product* object in a certain state in accordance with its defined life cycle, introduced in Fig. 3. The stored objects evolve over time changing their state by changing the records within the depicted table. The mapping of a state transition defined in the object life cycle to the implementation of the data extraction from the data source is established by a binding function. The implementation identifies the location of the timestamps of transition occurrence for each object at the model level. As in our case the data source is realized by a relational database, a database query stencil using, e.g., SQL as a query language, will describe the implementation. Listing 1.1 shows an example of a binding for an object state transition of the object *Product* from state *received* to state *in stock*.

```
bind ((Product , (received , in stock)) = {
SELECT in_stock FROM Product
WHERE Product.product_id = <product_id>
AND Product.received != null
AND Product.shipped != null;}
```

Listing 1.1. Definition of binding for an object state transition of the object *Product* from state *received* to state *in stock*

**Fig. 4.** Order process

For the order process, the company designed a process model, see Fig. 4, consisting of an initial activity *Analyze order* that uses an *received order* as input and transfers the *Order* to the state *confirmed* and may produce a data object *Type of shipment* in state *express*. After an order has been confirmed it is checked, whether the product request is in stock or not by the activity *Check stock* creating the data object *Product* either in state *in stock* or *not in stock*. Based on this outcome, a decision is made whether the product needs to be produced or products in stock fulfill the order. In case the product is not available, the raw material for the production is purchased (activity *Purchase raw material*) and the production plan is made (activity *Make production plan*). The activity *Purchase raw material* utilizes the data object *Product* and sets the state *purchased* to it once the required raw material is ordered. *Make production plan* has the data object *Product* as well and outputs the machines' setup (data object *Machines* in state *setup*). Once the purchased material is received, the Product is manufactured (activity *Manufacture product*) ending with the products being in stock again (data object *Product* in state *in stock*). When the produced order is in stock, the activity *Ship product* can be executed. This activity has a data object input set consisting of two elements requiring either the data object *Product* in state *in stock* or the data object *Product* in state *in stock* and the data object *Type of shipment* in state *express*. Once the activity is executed, the state *shipped* will be set to data object *Product* as well as to data object *Order*.

With the usage of the data objects in the process model described above, the information about the observed data state transitions can be used for process monitoring and progress prediction. How this is achieved will be discussed in the following section by application of our approach at the instance view.

4 Instance View Application by Use Case

In this section, we show the application of the introduced approach at the instance level using the order process defined in the previous section as example. First, we show how the identified bindings can be used for extraction of object state transition events from data sources at the IT Level. Afterwards, we discuss how they can be utilized to predict the behavior of the concrete process instance.

In the first step, we use the binding example introduced for the object *Product* in Listing 1.1 as a foundation for the definition of transition events. This binding defines exemplary the SQL stencil, which describes the access to the data set, reflecting the object state transition from state *received* to state *in stock*. This binding is part of the set of bindings, which describes object state transitions of the object *Product*: $\text{bind}(\text{Product}, (\text{received}, \text{in stock})) \in B_{\text{Product}}$. At the instance level, the defined binding can be used to access the data, which indicates the state transition of the instance of object *Product* with *product_id* = 856 described in Table 1. For this purpose, the SQL query stencil can be used replacing the placeholder $\langle \text{product_id} \rangle$ with the object's instance data - 856.

Further, the transition events can be extracted from the data sources using the object state transition event type $\mathcal{ET}_{\text{Product}} = (\text{Product}, B_{\text{Product}})$ under consideration of the appropriate bindings as described in the example above. One object state transition event for the state transition towards *in stock* may look as follows: $\mathcal{E}_1 = (\mathcal{ET}_{\text{Product}}, 0001; 2012-12-12 16:02; (856, \dots, 2012-12-12 16:02, 2012-12-11 10:40, 2012-12-11 10:51, 2012-12-12 15:13, \text{null}); \text{in stock})$. The event follows the structure introduced in the object state transition event definition (cf. Definition 3). The object is a record of the database at the point of time, when the event has occurred. The event *id* is generated.

Additionally, we observed the following object state transition events in sequence for the product 856. An event indicating the state *not in stock* was reached and from there the state *purchased* was reached. Another event indicates that the data object *product* was *received*. There is no event recognized for the shipping of the product yet, see Fig. 5. Analogously, we observed the state *confirmed* for the data object instance *order 411*. In the information system, there is no information available about the *type of shipment*. For machines, we cannot observe any state transition at all.

With the existing events, we can predict the progress of the order process instance for order 411. The techniques for correlation between instances of the order and the product data objects and the process instance are not part of this paper; they can be established by using approaches as presented in [15]. Because we cannot observe the state *received* of data object *order*, it is not possible to predict the enablement of activity *Analyze order*. However, we can monitor the state *accepted*, which is between the object states *received* and *confirmed* expected in the output set, but not modeled in the process model. Nonetheless, we can derive from that information, when the activity *Analyze order* began. The observation of the transition event of order 411 to state *confirmed* fulfills one of two sets of the output set ($\{\{(Order, confirmed)\}\}, \{(Order, confirmed), (Type of Shipment, express)\}\}$) of activity *Analyze order*, thus the activity is finished. Afterwards, we recognized a transition event for reaching the state *not in stock* of product 856. Since this is one valid output of activity *Check stock*, this activity is terminated as well. Based on the state of data object *Product*, the decision is made for producing the product. By observing the transition event to state *purchased* of instance 856 of data object *product*, it can be predicted that activity *Purchase raw material* is terminated. Analogously, this holds true for activity *Manufacture product* by recognizing the transition event for reaching the state *in stock* for product 856. However, we cannot predict the termination of the activity *Make production plan* from the available event information only. The control flow of the process model needs to be taken into account as well, so

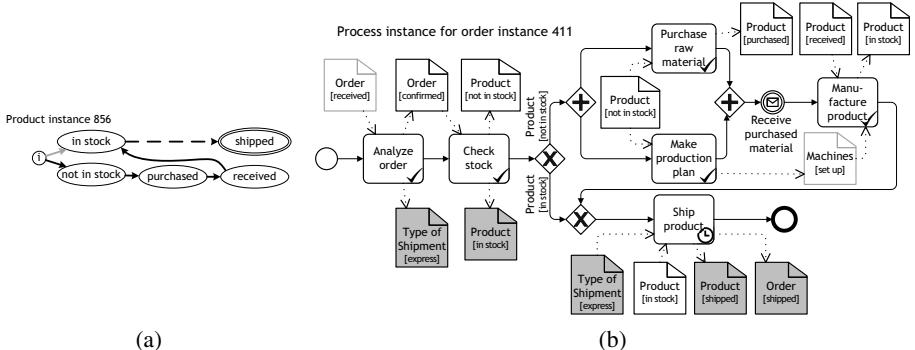


Fig. 5. Instance view of (a) the object life cycle for data object *Product* with id 856 and (b) the process instance for object *Order* with id 411. (a) shows for the product instance the current state as well as the historic ones based on the object state transition events generated from the information existing in the information systems. The specific product is currently in state *in stock* and reached it via states *not in stock*, *purchased*, and *received* (black arrows). It is expected to transition to state *shipped* some time in the future (dashed arrow). (b) shows the actual progress of the process instance of order 411 with activity *Ship product* being currently enabled. Thereby, all object state transition events relevant for this order are illustrated using the corresponding data objects: white colored data objects were observed in the respective state, gray colored data objects were not observed in the respective state yet, and gray bordered data objects cannot be observed in the respective state. Ticks visualize successful termination of activities, whereas the watch symbolizes an activity that is enabled respectively currently running.

that it can be assumed that once the activity *Manufacture product* is enabled, by observing object state transition event for product 856 and state *received*, the activity *Make production plan* must be finished before. With the transition event for product 856 and state *in stock*, the enablement of activity *Ship product* is denoted, because one set of the input set $\{\{(Product, instock)\}, \{(Product, instock), (Type of Shipment, express)\}\}$ of the activity is met. The activity is not finished so far since the output set is not fulfilled, e.g., $\{\{(Product, shipped), (Order, shipped)\}\}$. Both object state transition events are not observed yet. Even if one of those were already recorded, the activity cannot be set as terminated as the output set requires that both data objects, *Product* and *Order* are in state *shipped*. Thus, activity *Ship product* is enabled respectively running, indicated by the watch in Fig. 5.

Showing that the prediction of a process instance's progress, e.g., process instance for order 411, can be achieved by utilizing information about the state transitions of data objects illustrates the importance of the existence of object state transition events.

5 Related Work

In this section, we discuss the state of the art for process monitoring and analysis, event recognition, data modeling in business processes, and the correlation between process modeling artifacts and information systems.

As motivated, the presented approach targets on the enhancement of the event basis for business process intelligence (BPI). BPI aims to enable quality management for process executions by utilizing features like monitoring and analysis, prediction, control, and optimization [5]. A lot of literature discusses business process execution data capturing and storage [3, 5], however, most of them assume that every process step is observable and thus, the recorded event log is complete. [18] proposes a reference architecture for BPI, consisting of three layers, i.e., integration, function, and visualization. The presented approach targets in particular on the integration and the combination of information about data objects and the process execution. One application of BPI is process mining [2] that benefits from the presented approach, because more data about the process execution is made available. The events about the data object state transitions can be composed with the process execution events already existing to form a more fine-grained event log to enable existing process mining techniques.

The approach targets on the same vein as our previous works do where we present approaches for event recognition in especially manual executing process environments. In [6], an approach is presented that shows how event information out of information systems can be assigned to particular activities of a business process. [8] discusses how the recognition of events could be improved by utilizing process knowledge and external context data – data that exists independently of the business processes and their execution. The authors deal with correlation of events to each other but also the correlation to a process instance as well by applying common correlation techniques, like the algorithms for the determination of correlation sets based on event attributes introduced in [16].

Modeling data in the context of business processes received much attention such that most current process modeling notations support the modeling of data information [14] – especially for enacting process models. In this regard, requirements on data modeling capabilities are determined [9, 15]. Currently, the activity-centric (current standard way of modeling [15, 21]) and the object-centric modeling (cf., for instance, business artifacts [19] and the PHILharmomicFlows project [9]) paradigm are distinguished. Our approach introduced in this paper generally works with both paradigms, but several adjustments need to be undertaken to migrate it from one to the other paradigm. In this paper, we focused on activity-centric modeling because it is the current standard way of modeling and therefore the most used one such that our approach achieves highest impact. Furthermore, we assume the correlation of data object instances to process instances exists or can be ensured by the modeling approach [15].

Additionally, to support the design time part of approach, one may use a process model describing all actions and manipulations performed on data objects and synthesize the object life cycles from that model [4, 11]. The other way round, the complete process model can be extracted from object life cycles [10] potentially considering given compliance rules [12] to allow monitoring and analysis on all allowed aspects of a process. Data objects and their object life cycles to be used in process models may also be extracted from structured data sources as, for instance, ERP systems [20]. If the process model gets specified independently from the object life cycle, a conformance check can be applied to ensure that the process model only requests data manipulations allowed from the object life cycles of the data objects utilized in the process model [10, 13].

Various techniques from the area of information retrieval can be used to support automatic binding identification. We argue that in cases where data sources at the IT Level can be represented in form of an XML schema, e.g., relational databases, Web service requests and responses, structured event logs etc., approaches from the area of schema summarization [22] can be used to identify the relevant data structures, which represents the object state transition in the object life cycle. Additionally, schema matching techniques surveyed in [22] can be used to propose the matching of the object state transitions to the appropriate parts of the schema.

6 Conclusion

In this paper, we presented an approach to utilize object state transition events for reasoning about process progress and the enablement respectively termination of activities. Thereby, the transitions are linked to activities of the process model and contain, if they are observable, a binding, which links them into information systems, which hold information to recognize state changes for the observable transitions. If such state change is recognized for an object instance, a corresponding object state transition event is generated and then utilized for the mentioned reasoning. Combined with control flow based event recognition, this approach increases the number of events and therefore the log quality to be used in the field of business process intelligence for mining, monitoring, and analysis. In future work, we plan to also extract events from resource information to further improve the quality of the event log used for business process intelligence.

References

1. van der Aalst, W.M.P.: The Application of Petri Nets to Workflow Management. *Circuits Systems and Computers* 8, 21–66 (1998)
2. van der Aalst, W.M.P.: Process mining: Overview and Opportunities. *ACM Transactions on Management Information Systems (TMIS)* 3(2), 7:1–7:17 (2012)
3. Azvine, B., Cui, Z., Nauck, D., Majeed, B.: Real Time Business Intelligence for the Adaptive Enterprise. In: CEC/EEE, p. 29. IEEE (2006)
4. Eshuis, R., Van Gorp, P.: Synthesizing Object Life Cycles from Business Process Models. In: Atzeni, P., Cheung, D., Ram, S. (eds.) *ER 2012 Main Conference 2012. LNCS*, vol. 7532, pp. 307–320. Springer, Heidelberg (2012)
5. Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., Shan, M.: Business Process Intelligence. *Computers in Industry* 53(3), 321–343 (2004)
6. Herzberg, N., Kunze, M., Rogge-Solti, A.: Towards Process Evaluation in Non-automated Process Execution Environments. In: *ZEUS*, pp. 97–103 (2012)
7. Herzberg, N., Meyer, A.: Improving Process Monitoring and Progress Prediction with Data State Transition Events. In: *ZEUS* (2013)
8. Herzberg, N., Meyer, A., Weske, M.: An Event Processing Platform for Business Process Management. In: Gaševic, D., Hatala, M., Motahari Nezhad, H.R., Reichert, M. (eds.) *Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference*, pp. 107–116. IEEE Computer Press (September 2013)
9. Künzle, V., Reichert, M.: PHILharmonicFlows: Towards a Framework for Object-aware Process Management. *Journal of Software Maintenance* 23(4), 205–244 (2011)

10. Küster, J.M., Ryndina, K., Gall, H.C.: Generation of Business Process Models for Object Life Cycle Compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 165–181. Springer, Heidelberg (2007)
11. Liu, R., Wu, F.Y., Kumaran, S.: Transforming Activity-Centric Business Process Models into Information-Centric Models for SOA Solutions. *J. Database Manag.* 21(4), 14–34 (2010)
12. Lohmann, N.: Compliance by design for artifact-centric business processes. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 99–115. Springer, Heidelberg (2011)
13. Meyer, A., Polyvyanyy, A., Weske, M.: Weak Conformance of Process Models with respect to Data Objects. In: ZEUS, pp. 74–80 (2012)
14. Meyer, A., Smirnov, S., Weske, M.: Data in Business Processes. *EMISA Forum* 31(3), 5–31 (2011)
15. Meyer, A., Pufahl, L., Fahland, D., Weske, M.: Modeling and Enacting Complex Data Dependencies in Business Processes. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 171–186. Springer, Heidelberg (2013)
16. Motahari-Nezhad, H.R., Saint-Paul, R., Casati, F., Benatallah, B.: Event correlation for process discovery from web service interaction logs. *VLDB Journal* 20(3), 417–444 (2011)
17. Murata, T.: Petri nets: Properties, Analysis and Applications. *Proceedings of the IEEE* 77(4), 541–580 (1989)
18. Mutschler, B., Bumiller, J., Reichert, M.: An Approach to Quantify the Costs of Business Process Intelligence. In: EMISA, pp. 152–163 (2005)
19. Nigam, A., Caswell, N.S.: Business artifacts: An approach to operational specification. *IBM Systems Journal* 42(3), 428–445 (2003)
20. Nooijen, E.H.J., van Dongen, B.F., Fahland, D.: Automatic Discovery of Data-Centric and Artifact-Centric Processes. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBP, vol. 132, pp. 316–327. Springer, Heidelberg (2013)
21. OMG: Business Process Model and Notation (BPMN), Version 2.0 (2011)
22. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. *VLDB Journal* 10 (2001)
23. Weske, M.: Business Process Management: Concepts, Languages, Architectures, 2nd edn. Springer (2012)
24. Yang, X., Procopiuc, C.M., Srivastava, D.: Summarizing Relational Databases. *VLDB Endowment* 2(1), 634–645 (2009)

A Conceptual Model of Intended Learning Outcomes Supporting Curriculum Development

Preecha Tangworakitthaworn, Lester Gilbert, and Gary B. Wills

Department of Electronics and Computer Science,
Faculty of Physical Sciences and Engineering, University of Southampton
United Kingdom
`{pt2e10, lg3, gbw}@ecs.soton.ac.uk`

Abstract. Designing and developing a logical structure of intended learning outcomes (ILOs) by using a diagrammatic technique, is a challenge for instructional designers in their systematic design and development of curriculum. In this paper, an ILO diagram – a novel conceptual model for curriculum development, is proposed. The modelling of ILOs and its components are introduced as well as the relationships and constraints of the ILO diagram are proposed. Finally, a scenario for applying the proposed ILO diagram in education is elaborated upon.

Keywords: Curriculum Modelling, Curriculum Development, Conceptual Model, Instructional Design, Visualization, Intended Learning Outcome, ILO.

1 Introduction

Recently, distance learning (or e-learning) plays a crucial role as an application domain of Software Engineering [7]. In a general software development process, a data model called an entity-relationship diagram (or ERD) [8] is a renowned conceptual model for database design and development. Although, an ERD embodies some semantic information about the business requirements, a conceptual model used by instructional designers in their systematic design of curriculum development should incorporate specific information about educational contexts and facilitate every stakeholder in educational domain.

In order to design a lesson, course, or programme that serves both learners and instructors, an outcome-based education approach is required in order to focus on what learners will be able to do after they are taught [1]. Moreover, focusing on the learning goals is the main characteristic of intended learning outcomes (or ILOs) that leads to the powerful design of an educational programme and curriculum [6].

This paper presents a conceptual model for curriculum development that allows instructional designers to model a logical structure of learning content and learning materials in which the subject matters and their relationships are integrated with the capabilities to be learned. The ILO structural design is proposed to support not only instructional designers to design and develop courses of study systematically, but also instructors and learners in undertaking teaching and learning activities.

The following sections discuss four aspects of the research reported in this paper. Section 2 proposes challenges in conceptual modelling for curriculum development. In section 3 an approach of a conceptual model for curriculum development is proposed through the design and development of an ILO diagram, whilst in section 4 a scenario of applying the proposed ILO approach in academia is introduced. Finally, in section 5 some conclusions are drawn.

2 Challenges in Conceptual Modelling for Curriculum Development

In this research, there are two main challenges of designing a conceptual model for curriculum development. The first challenge is the visualization of learning content and learning materials. The intended learning outcome plays a crucial role as the infrastructure of this research to be the representative unit of the learned capability and learning materials. Traditionally, all ILOs of a specific course of study are expressed as plain text or unstructured documents. Learning by referring to unstructured ILOs may lead to an inability to understand the whole structure of the course content and learning materials. Thus, the aim of this research is to contribute a logical structure of ILOs by using a diagrammatic technique as a tool for curriculum development.

The second challenge is the design and development of a logical structure of intended learning outcomes, in which subject matters and their relationships are integrated with the capabilities to be learned. A main feature of the proposed conceptual model is that it exposes what learners have expected to achieve. The proposed approach focuses on educational activities defined in terms of what the learner should achieve by the end of the lesson, course, or programme.

3 A Conceptual Model for Curriculum Development

This research proposes a novel conceptual model for curriculum development to support not only instructional designers in their systematic design and development of courses of study, but also learners and instructors in undertaking learning and teaching activities. In order to develop the proposed conceptual model, the framework identifies an outcome-based learning expression through intended learning outcomes. The idea of using intended learning outcomes can guide the instructional designers to plan the learning goals for the course modules as well as to initiate the learning objectives which will be officially declared to support the curriculum development.

3.1 Intended Learning Outcome

A statement, the intended learning outcome (ILO), is the planned learning outcome which expresses the students' ability to be able to perform learning activities by the end of the course modules [9, 13, 14]. The ILO has commonly planned and desired before providing the learning tasks to the learners [2]. Traditionally, an ILO begins, "by the end of the course, the learner will be able to..." X and Y, where X is capability and Y is subject matter content [12]. An ILO is normally expressed in terms of the plain text that defines to identify the learning objectives of the course of study [12, 14].

3.2 Matching Learner and Instructor ILOs

The instructor and the learner share the pedagogical content of the instructor's goals and the learner's goals, instructor's knowledge and learner's knowledge, and the instructing activities and learning activities. Fig.1 illustrates the matching perspective of the ILOs.

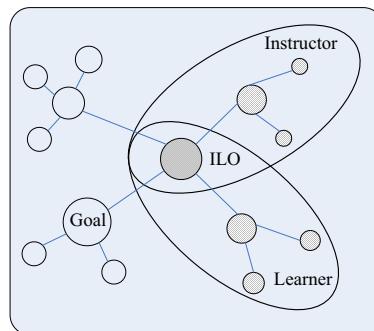


Fig. 1. Matching Perspective of ILOs

Besides the instructor's and learner's views, there is a matching perspective that normally occurs during the course of study. This is because the teacher and the learner share similar goals of the pedagogical activities: teaching and learning activities. It is their joint intention to gain an understanding of the subject matter content (also called learning materials) which is the ideal of the pedagogical activities. Hence, the shared goals are determined to be the indication leading to the improvement of the learned capabilities.

3.3 Designing ILOs' Components

In this research, the ILOs can be decomposed into two main components, namely, capability, and subject matter content. The following subsections discuss the details of these two components.

Capability

The capability component deals with the learner's ability to perform the learning activities. The capability of an ILO refers to a verb designating the learned capability. In this research, the learned capability verb (LCV) has been denoted as the action word which is expressed the linguistic element of ILOs. In addition, Bloom's taxonomy of cognitive domain [4] has been adopted to represent the capability component of ILOs. For instance, an ILO may state, "by the end of the course, student will be able to *design* the entity relationship diagram"; the learned capability verb of this example is "*design*".

In this research, six levels of Bloom's cognitive domain [4] form the basis of a cognitive hierarchy. Table 1 shows the examples of LCVs classified for each level of the revised cognitive taxonomy proposed by Anderson & Krathwohl [3].

Table 1. Examples of LCVs for each level of Cognitive Taxonomy [3, 4, 14]

Revised Cognitive Taxonomy	LCVs
Create	assemble, categorise, create, design, establish, formulate, generalise, generate, integrate, organise, produce
Evaluate	appraise, argue, assess, check, contrast, criticise, critique, evaluate, judge, justify, measure, resolve
Analyse	analyse, attribute, break down, categorise, classify, compare, differentiate, distinguish, examine, organize, test
Apply	apply, assess, change, construct, demonstrate, develop, execute, experiment, implement, operate, use
Understand	associate, change, clarify, compare, describe, exemplify, explain, express, identify, indicate, infer, interpret, report, summarise
Remember	collect, define, describe, enumerate, label, list, name, order, present, recognise, recall, state

Although Anderson & Krathwohl have proposed an updated version [3], this research has adopted Bloom's original taxonomy as the fundamental part of the proposed conceptual model and ILOs' logical structure (see Fig 3).

Subject Matter Content

The subject matter content (SMC) identifies the learning content or learning materials of the course of study. In this research, based on the component display theory proposed by Merrill [15, 16], there are four categories of SMC, namely, fact, concept, procedure, and principle. A fact is defined as two associated parts of information, such as, a specific name and a date, an event and the particular name of a place, etc. A concept is a concrete or abstract item with certain characteristics, such as, a human being is a primate with a bipedal gait, etc. A procedure (or process) is a set of steps for accomplishing some objectives, such as, a computer program, a recipe for cooking Thai food, etc. Finally, a principle is a cause-and-effect relationship that predicts outcomes, such as, road accidents occur because of slippery roads, apples fall because of the gravity, etc. Table 2 illustrates examples of each category of SMC.

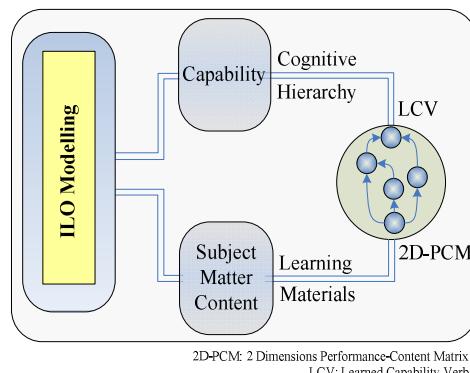
Table 2. Examples of each SMC category

SMC Category	Examples
Fact	- Who is the current king of Thailand? - The colour of the gold is yellow. - What is the official URL of University of Southampton?
Concept	- Identify characteristics of human error. - What are the main properties of a hybrid car? - A student record comprises StdName, BDate, Age, etc.
Procedure	- Specify the steps required to prepare and manipulate the data for Data Mining. - Please explain how to cook Thai green curry. - Develop the web application by using Java.
Principle	- Identify the causal evidence of the road accident. - Given a basic equation A, solve the problem B. - State reasons why we need to change the password of internet account.

Each category of SMC has been identified to materialise the learning contents embodied in an ILO expression. Following Merrill [15], we define the two-dimensional performance content matrix as 2D-PCM, using Merrill's classification scheme [16] to represent the relationship between learner's performance and learning materials (defined in terms of subject matter content). The first dimension is the performance which comprises three types: find, use and know (or remember). The second dimension is subject matter content which comprises four types: fact, concept, procedure, and principle. Thus, ten relationships of 2D-PCM are defined, namely, know-fact, know-concept, know-procedure, know-principle, use-concept, use-procedure, use-principle, find-concept, find-procedure, and find-principle.

3.4 Visualising ILOs through an ILO Diagram

In order to visualise all ILOs of a specific course, lesson, or programme through a logical structure, two main components of ILOs are determined as depicted in Fig 2.

**Fig. 2.** The ILO Modelling

First, the capability component refers to the action or activity of learners in performing learning tasks. Six categories of Bloom's cognitive taxonomy forming the cognitive hierarchy are adopted to express the capability of ILO by using the learned capability verb, LCV. Second, the subject matter component refers to the learning material which is represented by using the 2D-PCM. Fig.2 illustrates the process of the ILO modelling contributed in this research.

A traditional ILO statement expressed in plain text can be formed as an ILO node of an ILO diagram. Structurally, each ILO node consists of four elements, namely, ILO number, 2D-PCM, LCV, and SMC. The ILO number identifies the node in an ILO diagram. The 2D-PCM represents the classification of the node within the performance/content matrix. The SMC represents the subject matter content of the ILO, and it is used to show relationships between ILOs with matching or similar SMCs. The LCV of each node is used in two ways. Firstly, it is mapped to the cognitive hierarchy as illustrated in Fig.3. Secondly, and more significantly, enabling ILO is related to higher-level ILOs through consideration of the LCV (called LCV mapping).

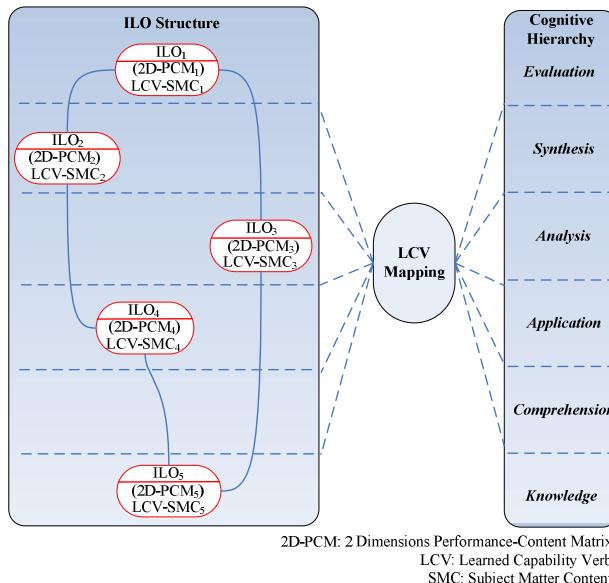


Fig. 3. The LCV Mapping Scheme [17]

In principle, the ILO diagram can be firstly designed by augmenting the ILO structure with the cognitive hierarchy based on Bloom's taxonomy, but later on the design and development of the ILO diagram can be applied to other taxonomies, such as Gagné's hierarchy of learned capabilities [11], or Merrill's level of performance [16].

3.5 ILO Relationships

In this research, a relationship of one ILO to another represents either a partial or a whole part that shares some elements (i.e., LCV, SMC, or both LCV and SMC) in common. It is important to note that two basic elements of an ILO node: LCV, and SMC, play important roles in relationship design because these two elements are the representative units of the basic component of ILOs. Thus, there are two types of ILO relationship, namely, partial, and whole part. The partial represents the fundamental structure of a basic component that holds either LCV or SMC; hence, this relationship is named as the *principal relationship*. Whilst, the whole part is determined by both LCV and SMC elements, so the name of the relationship is *composite relationship*.

There are three types of the principal relationship (see Table 3).

Table 3. Three Types of Principal Relationship

Type	Notation	Description
Capability Relationship		when LCV relates to enabling LCV
Topic Relationship		when SMC relates to SMC
Inheritance Relationship		when SMC relates to superclass SMC

First, the capability relationship represents the link between learned capability verbs, LCV. The value of an ILO diagram is given when ILOs which enable higher-level ILOs are identified. The result supports learning paths and learner positioning within a learning domain. In constructing the ILO diagram, enabling ILOs are identified by their LCVs being enablers of other LCVs, called "eLCV's" or "enabling LCV's". For example, "*develop*" is an enabling LCV of "*test*". This is because "*develop*" is a prerequisite capability of "*test*" in the cognitive taxonomy. The ILO diagram notation for the capability relationship is a solid arrowhead placed near the centre of relationship line.

Second, the topic relationship represents the link between subject matter contents, SMC. A group of ILOs share a common topic if it has a common SMC, resulting in a topic relationship. For example, "*describe DFD*" shares a common SMC with "*change DFD*". The ILO diagram notation for the topic relationship is a simple line.

Third, the inheritance relationship represents the link between superclass/subclass SMC within subject matter hierarchy. The SMCs of two ILOs can have an inheritance relationship if one SMC refers to the superclass of the other (sSMC). This relationship is based on the class hierarchy of an object-oriented UML class diagram [18]. For instance, a data warehouse SMC may be identified as the superclass SMC of a data mart. The ILO diagram notation for the inheritance relationship is a line with an open arrowhead placed at the superclass.

In addition, the composite relationship is determined by combining two components of ILOs, i.e., capability, and subject matter content. It occurs when two ILO nodes have two principal relationships. There are two types of composite relationship expressed in Table 4.

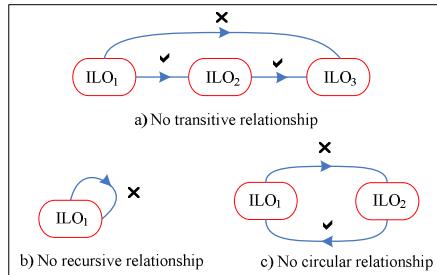
Table 4. Two Types of Composite Relationship

Notation	Description
	a) when capability and inheritance relationships occur with opposite orientation.
	b) when capability and inheritance relationships occur with the same orientation.

The composite relationship has determined when LCV (or eLCV) is linked to LCV (or eLCV) and SMC (or sSMC) is related to SMC (or sSMC). For instance, a composite relationship connects "*design simple ERD*" with "*evaluate logical model*", when "*design*" is an enabling LCV of "*evaluate*" and "*logical model*" is a superclass SMC of "*ERD*".

3.6 ILO Relationship Constraints

Although it can be useful to apparently design the ILO relationships, the conceptual model of ILOs should contribute modelling construction for explicitly supporting the pedagogical activities. Based on the educational purposes of the course design [12], we propose that there are three constraints of the ILO relationship which are illustrated in Fig 4.

**Fig. 4.** Three constraints of ILO relationship

First, no transitive relationship plays important role in designing the ILO diagram. Whenever ILO_1 is related to ILO_2 and ILO_2 is related to ILO_3 , then the relationship of ILO_1 is not obviously transferred to ILO_3 . This is because not only the capability part of the ILO cannot be conveyed, but also the subject matter content part cannot be transmitted from ILO_1 to ILO_3 . For example, a learner can evaluate the ER model if he/she can previously identify the business rules and then draw the basic ER model, but he/she cannot evaluate it without drawing the ER model completely.

Second, the ILO diagram should not construct the recursive structure, when a single ILO node is related to itself. Referring to the inheritance relationship of the ILO, each ILO node is instantiated from the competency class [12]. This means that when the ILO has been referred to the instance level of the class, it cannot hold the recursive relationship. The reason is that when the prerequisite behaviour has been furnished to the ILO, it is not a self-contained behaviour.

Third, the last constraint is no circular relationship. The principle of educational objective abstractly reveals that if ILO_1 is a prerequisite of ILO_2 , then ILO_2 cannot be the prerequisite of ILO_1 simultaneously. This leads to preventing the recursive relationship of the ILO diagram.

3.7 An ILO Diagram

In order to demonstrate how to apply the conceptual model of ILOs (ILO diagram) in designing a course of study, we consider the available published course document of IT curriculum proposed by ACM Special Interest Group on IT Education that conforms to the emerging accreditation standards for IT program [10]. In this study, the chosen course is Information Management (IM4) Data Modelling unit.

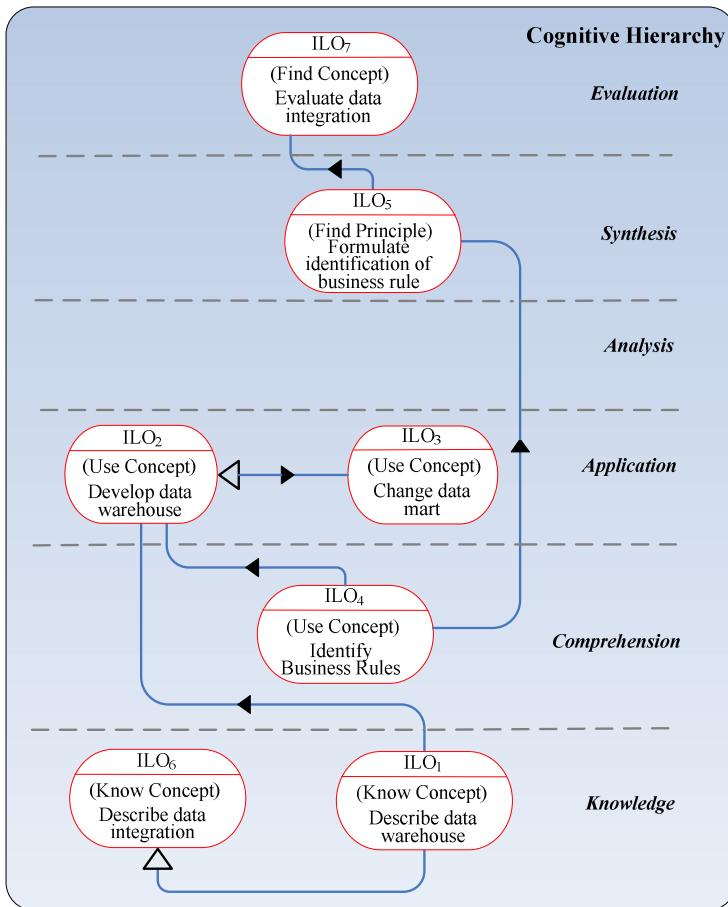


Fig. 5. An ILO Diagram of the IM4 Data Modelling Module

We consider seven intended learning outcomes of the IM4 data modelling module to represent as seven ILO nodes. We analyse and assign the suitable level of cognitive hierarchy by referring to the LCV mapping mechanism as well as the ILO relationships and constraints have been assigned and determined to each pair of the ILO nodes. Then we can obtain the ILO diagram as illustrated in Fig. 5.

A case study reviewed above represents that the proposed ILO diagram has been introduced to completely visualise the course structure of the data modelling module in the IT curriculum.

4 A Scenario for Applying the Proposed Approach in Education

We elaborate the initialisation steps, in which the proposed approach can promote the state-of-the-art conceptual model in education. In order to systematise the processes of course design, course development, and course delivery, we show in Fig.6 as a UML use case diagram.

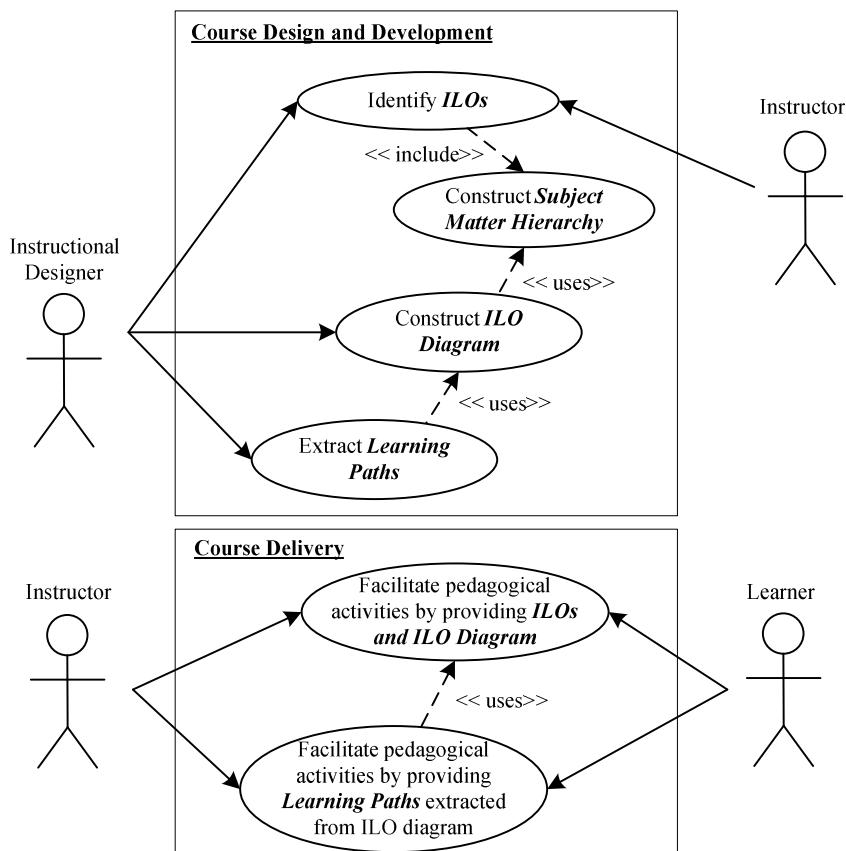
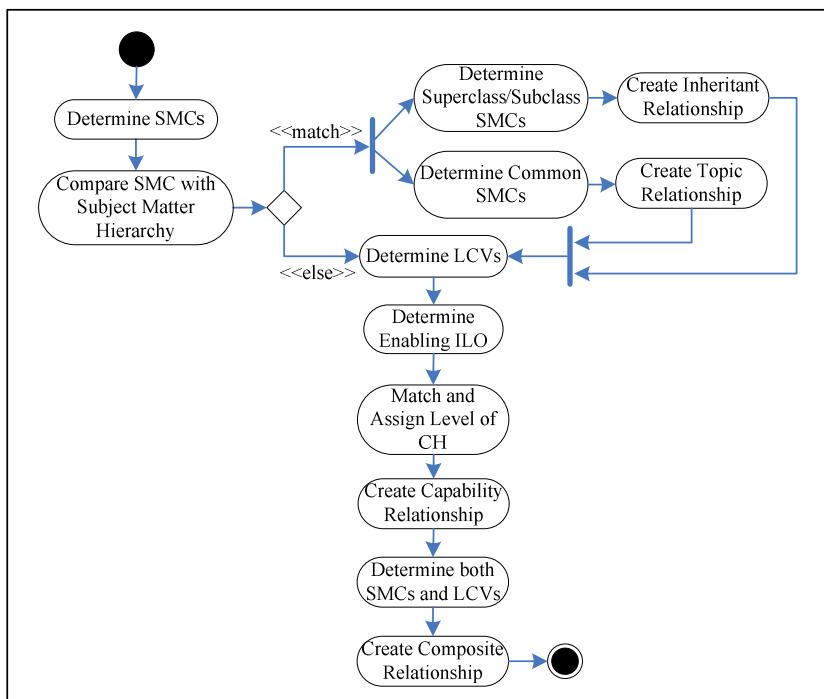


Fig. 6. Use Case Diagram for Course Design and Development and Course Delivery

In the first phase which is the course design and development, the first process is to identify all ILOs of a course of study. An instructor provides the educational requirements by declaring as the pedagogical goals expressed in unstructured ILOs format (or plain text). In order to assign the structured format for all ILOs, instructional designer comes up with the ILOs' components declaration by assigning four elements for each ILO (i.e., ILO Id, LCV, SMC, and 2D-PCM) and forming the ILO nodes. Then, the subject matter hierarchy has been designed and created in order to construct the super-class/subclass SMCs. Next, the ILO diagram construction process is performed and the detailed algorithm is illustrated by using the activity diagram depicted in Fig 7.



LCV: Learned Capability Verb
 SMC: Subject Matter Content
 CH: Cognitive Hierarchy

Fig. 7. Activity Diagram for Constructing an ILO diagram

All activities of the ILO diagram construction illustrated in Fig 7 show that the modelling construction can be achieved by 1) determining and comparing SMCs with the subject matter hierarchy and formulating the inheritance and topic relationships; 2) determining LCVs, enabling ILO nodes, matching LCVs and assigning the suitable level of cognitive hierarchy for each ILO node, and then formulating the capability relationship; 3) determining both SMCs and LCVs to formulate the composite relationship.

As a consequence, in the final process of the course design and development phase (see Fig 6), we propose the learning paths extraction which generates the sequences of pedagogical activities. Instructional designer can extract the learning paths visualised via an ILO diagram. The results can be utilised to suggest the appropriate direction to learners in order to perform the suitable learning activities to achieve their learning goals.

In the second phase which is the course delivery, two main processes are introduced as additional facilitators for teaching and learning. The first process is to provide all ILOs and an ILO diagram to support pedagogical activities, whilst the second process is to suggest the learning paths extracted from an ILO diagram to the class room (or learning environment). Instructors and learners can refer to the suggested learning paths in executing the learning materials in order to achieve the desired pedagogical goals.

5 Conclusions

In order to pioneer courses of study which should consider all stakeholders in education, an approach of intended learning outcome (ILO) has been introduced to indicate what learners will be able to do by the end of the course of study. Traditionally, an ILO statement expresses as plain text or unstructured document. The research aims to contribute the design of a logical structure of ILOs by introducing a diagrammatic technique that incorporates information about educational contexts and learners' capabilities.

In this paper, a novel conceptual model called an ILO diagram was introduced to visualise a logical structure of ILOs through the design of ILO relationships and constraints in order to support the curriculum development. In addition, facilitating ILOs and an ILO diagram in course design, course development, and course delivery was proposed and illustrated.

References

1. Allan, J.: Learning Outcomes in Higher Education. *Studies in Higher Education* 21(1), 93–108 (1996)
2. Anderson, H.M., Moor, D.L., Anaya, G., Bird, E.: Student Learning Outcomes Assessment: A Component of Program Assessment. *American Journal of Pharmaceutical Education* 69(2), 256–268 (2005)
3. Anderson, L.W., Krathwohl, D.R.: *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Addison, Wesley, Longman, New York (2001)
4. Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R.: *Taxonomy of Educational Objectives: Handbook 1 Cognitive Domain*. David McKay, New York (1956)
5. Bloom, B.S., Madaus, G.F., Hastings, J.T.: *Evaluation to Improve Learning*. McGraw-Hill, New York (1981)

6. Bouslama, F., Lansari, A., Al-Rawi, A., Abonamah, A.A.: A Novel Outcome-Based Educational Model and its Effect on Student Learning, Curriculum Development, and Assessment. *Journal of Information Technology Education* 2, 203–214 (2003)
7. Caeiro-Rodríguez, M., Llamas-Nistal, M., Anido-Rifón, L.: Modeling Group-Based Education. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 96–111. Springer, Heidelberg (2005)
8. Chen, P.: The Entity-Relationship Diagram: Toward a Unified View of Data. *ACM Trans. on Database Systems* 1(1), 9–36 (1976)
9. Dodridge, M.: Learning Outcomes and Their Assessment in Higher Education. *Engineering Science and Education Journal*, 161–168 (1999)
10. Ekstrom, J.J., Gorka, S., Kamali, R., Lawzon, E., Lunt, B., Miller, J., Reichgelt, H.: The Information Technology Model Curriculum. *Journal of Information Technology Education* 5, 343–361 (2006)
11. Gagné, R.M.: *The Conditions of Learning*. Holt, Rinehart and Winston, New York (1965)
12. Gilbert, L., Gale, V.: *Principles of E-Learning Systems Engineering*. Chandos Publishing, Oxford (2008)
13. Harden, R.M., Crosby, J.R., Davis, M.H.: AMEE Guide No.14: Outcome-Based Education: Part1 – An Introduction to Outcome-Based Education. *Medical Teacher* 21(1), 7–14 (1999)
14. Kennedy, D., Hyland, A., Ryan, N.: *Writing and Using Learning Outcomes: A Practical Guide*. University College Cork (2007)
15. Merrill, M.D.: Content and Instructional Analysis for Cognitive Transfer Tasks. *Audio Visual Communications Review* 21, 109–125 (1973)
16. Merrill, M.D.: The Descriptive Component Display Theory. In: Twitchell, D.G. (ed.) *Instructional Design Theory*, pp. 111–157. Educational Technology Publications, New Jersey (1994)
17. Tangworakitthaworn, P., Gilbert, L., Wills, G.B.: Designing and Diagramming an Intended Learning Outcome Structure: A Case Study from the Instructors' Perspective. In: The 13th IEEE International Conference on Advanced Learning Technologies (ICALT 2013), Beijing, China (2013)
18. UML Revision Taskforce.: OMG Unified Modeling Language (OMG UML): Superstructure Version 2.3,
<http://www.omg.org/spec/UML/2.3/Superstructure/PDF>
(accessed online April 2, 2013)

Cost-Informed Operational Process Support

Moe T. Wynn¹, Hajo A. Reijers^{2,3}, Michael Adams¹, Chun Ouyang¹,
Arthur H.M. ter Hofstede^{1,2}, Wil M.P. van der Aalst^{2,1},
Michael Rosemann¹, and Zahirul Hoque⁴

¹ Queensland University of Technology, Brisbane, Australia

{m.wynn,mj.adams,c.ouyang,a.terhofstede,m.rosemann}@qut.edu.au

² Eindhoven University of Technology, Eindhoven, The Netherlands

{h.a.reijers,w.m.p.v.d.aalst}@tue.nl

³ Perceptive Software, Apeldoorn, The Netherlands

hajo.reijers@perceptivesoftware.com

⁴ La Trobe University, Melbourne, Australia

z.hoque@latrobe.edu.au

Abstract. The ability to steer business operations in alignment with the true origins of costs, and to be informed about this on a *real-time basis*, allows businesses to increase profitability. In most organisations however, high-level cost-based managerial decisions are still being made separately from process-related operational decisions. In this paper, we describe how *process-related decisions at the operational level can be guided by cost considerations* and how these *cost-informed decision rules can be supported by a workflow management system*. The paper presents the conceptual framework together with data requirements and technical challenges that need to be addressed to realise cost-informed workflow execution. The feasibility of our approach is demonstrated using a prototype implementation in the YAWL workflow environment.

Keywords: Cost-Informed Process Enactment, Business Process Management, Workflow Management, Process Modelling, Prototype.

1 Introduction

Organisations are eager to implement cost-based considerations in their day-to-day operations. In most organisations, however, tying cost considerations to process-related decisions forms a challenge. Our observation is that most Workflow Management Systems (WfMSs) offer no support for cost considerations beyond the use of generic attributes (e.g. FileNet Business Process Manager) or some basic cost recognition and reporting (e.g. TIBCO Staffware Process Suite). Detailed cost information is typically not available at runtime and, as a result, cost information is not used for monitoring or for operational decision support.

Our motivation for this paper is to provide a conceptual framework to enable WfMSs¹ to achieve a higher level of support for cost-informed operational

¹ In the remainder, we use the term WfMS to refer to all process-aware information systems, including Business Process Management Systems (BPMSS).

decisions. More specifically, such a cost-aware WfMS is able to record historical cost information and makes use of it for (real-time cost) monitoring and escalation purposes, as well as supporting simulation and cost prediction capabilities. Ideally, it can also support process improvement decisions based on cost considerations, such as determining cost profiles of different processes/process variants and using them for selection/redesign purposes. To this end, we propose methods for the capture of cost-based decision rules for process, activity and resource selections within business processes and how automated support could be provided for cost-informed process enactment within a WfMS.

It is worth noting that cost is traditionally considered as one of many non-functional requirements (NFR) for a software system or service in the same manner as maintainability, usability, reliability, traceability, quality or safety [2]. However, the cost perspective has a very close and direct link with BPM/WfM, much more so than most other NFRs. First of all, consider that cost is relevant from the viewpoint of individual activities, resources, and entire processes – all of which are in scope for a WfMS. This *versatility* typically does not hold for many other NFRs. Quality, for example, is relevant in the context of a whole process, but not necessarily for a single activity; usability can be tied to a single activity, but not to resources; reliability may be relevant for a single activity, but is too fine-grained for cross-functional processes. Secondly, when we refer to the *dynamic* nature of cost we mean that it is relevant for both design and run time decisions. This aspect differs from NFRs such as maintainability and usability, which are important concerns at design time, but out of scope for operational decision making. Again, both the design and run time perspectives are in scope for a WfMS. In summary, a WfMS is a natural platform to manage cost concerns since it connects the many levels of cost interests and allows for implementing cost-informed design and operational decisions. Hence, we propose a conceptual framework which is tailored towards incorporating the cost perspective within a WfMS with the specific goal of supporting cost-informed operational decisions.

2 A Framework for Cost-Informed Decisions

Different types of actions can be performed by a WfMS or by a resource interacting with a WfMS to support *cost-informed decision making during process execution*. We propose that in addition to the ability to specify cost-informed control flow definitions and resource allocation rules *at design time*, a cost-informed WfMS should provide support for system-based decisions and system-supported user decisions *at runtime*. Figure 1 depicts our conceptual framework which describes 1) *data input*, i.e. the information requirements to enact actions that can be undertaken by or with a WfMS, 2) the *actions* that can be taken on the levels of process, activity, and resource (work distribution), and 3) the *cost-informed support* that is delivered, either through decisions by the WfMS itself or people using its support.

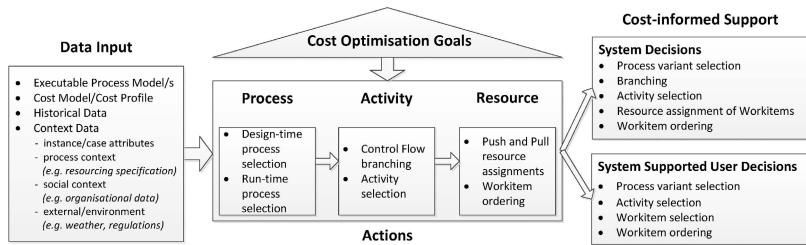


Fig. 1. A framework supporting cost-informed process execution with a WfMS

2.1 Data Input

A number of key objects need to be provided to a WfMS as *data inputs* to support cost-informed actions. In addition to an *executable process model*, we need access to a *cost model* that can be associated with different elements within a process model. Cost data could be as simple or as complex as an organisation requires it to be. For instance, it could be a variable cost that describes the hourly rate of a resource, but it could also be a dynamic scheme that ties overhead costs to each case depending on seasonal factors. Cost information, together with *historical data* as stored in a so-called *process log* regarding past executions, can be used to determine the cost of process executions as illustrated in our earlier work [14]. Since a business process is always executed in a particular context, we also adopt the four levels of *context data* described in [11]: case attributes, process context, social context, and the environment.

2.2 Actions

All cost-informed actions are based on the data inputs that we discussed on the one hand, while they are governed by the strategic considerations within an organisation on the other. We refer to these as *cost optimisation goals*. Typical examples are: cost minimisation, cost overrun prevention, profit maximisation, incorporation of opportunity cost, etc. The concrete cost-informed actions supported by a WfMS, informed by data input and governed by cost optimisation goals, can then be classified into three levels: process, activity, and resource.

- **Process.** The *process* level is concerned with carrying out *process selection* based on cost information of processes or process variants at design time or at runtime. This may involve the selection among different processes or selection among different process variants (which are created from individual processes during the execution phase). It should also be possible to assign a (whole) process or process variant to a certain resource team for execution (i.e. outsourcing) based on the cost profile.
- **Activity.** For cases that have been started under the control of a WMFS, it is necessary to decide at certain points about the *activity* (or activities) to be

executed next. In its coordination capability, a WfMS may decide on which workitems are enabled in a specific case, based on the branching conditions specified in the control-flow of the underlying process model. A WfMS could also start, skip, and cancel a workitem, among other actions, based on that cost information.

- **Resource.** After a workitem has been enabled, further choices related to distributing work to *resources* become possible. For workitems that need to be carried out by a user, both “push” and “pull” patterns of activity-resource assignment [8] should be supported.

Figure 2 shows possible cost-based decision points within the lifecycle of a workitem (transitions that can be cost-informed are depicted using bold arrows). After a workitem is created, the system can offer the workitem to one or more resources for execution (which is depicted as “S:offer_s” and “S:offer_m” decisions). An additional “C:selection” annotation indicates that it is possible for this system decision to be cost-informed. i.e. a resource could be selected based on its cost characteristics. After a workitem is started by a resource, it can still be suspended/resumed or cancelled by a resource. The “R:suspend”, “R:resume”, and “R:cancel” transitions reflect these possibilities and similarly the “C:decision” annotations in these transitions indicate that these user decisions can be guided by cost information. When more than one workitem is assigned to a resource and/or when a workitem is offered to multiple resources, a WfMS can provide support for the prioritisation of workitems based on cost information using the concept of cost-based orderings, i.e., “C:ordering”.

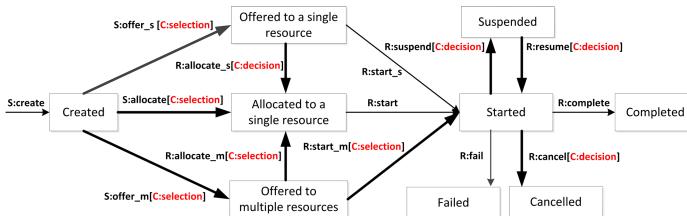


Fig. 2. Lifecycle of a workitem (based on [8]) – enriched with potential cost-based rules for system decisions and system-supported user decisions

2.3 Cost-Informed Support

As we mentioned, our framework identifies the two types of *cost-informed support* that result from the discussed ingredients: *systems decisions*, which can be taken by the WfMS itself, and *system-supported user decisions*, which are taken by resources on the basis of information provided by the WfMS. For instance, it is possible for the WfMS to make an automated selection of the process variant based on its cost profile and context information. Alternatively, the WfMS can provide the resource with cost profiles of different process variants and the resource can make the selection. This is also true for decisions on which activities to execute. The WfMS can either make a cost-informed decision based on

a pre-defined business rule to enable/start an activity or allow the resource to start/skip/suspend/resume/cancel a particular activity based on cost information. Decisions on which paths to choose in a process are exclusively taken care of by the WfMS using predefined cost-informed business rules. Workitems can be assigned by the WfMS or can be selected by a resource based on their cost (historical or predicted values).

2.4 Technical Challenges

For a WfMS to be capable of cost-informed enactment, execution and support across the three levels (process, activity and resource), the following key criteria would need to be satisfied:

1. *Association of cost data and cost-based rules with a process/workflow.* This support is required prior to the execution phase. Relevant cost rates for different process elements such as activities and resources must be specified in advance. Some values would include salary and incidental costs for human resources, the costs of materials required, fixed costs associated with activity enactments, rentals, depreciation, and so on.
2. *Runtime calculation of the cost of execution of each process instance and its component activity instances.* Such calculations may be based on time, usage, measurement, invocation, a fixed cost, or a combination of the above.
3. *Logging and analysis of cost data.* The ability to archive all calculated costs for each process instance (incorporated into the process event logs) and to perform extrapolated calculations over archived data.
4. *Support for cost-informed decisions.* The ability to use the calculated cost for the current process instance, and/or those of all previous instances of the process, to make cost-informed decisions.

3 Realisation

We have developed a prototype implementation within the YAWL workflow environment [9]. YAWL was chosen as the implementation platform because it is built on an expressive workflow language that provides extensive support for identified workflow and resource patterns, together with a formal semantics.

A new YAWL custom service, known as the *Cost Service*, is responsible for performing the required cost calculations by applying the data to the relevant cost model components; and for storing all interactions and results in its process logs. The workflow engine and other interested services such as the *Resource Service*, which manages all resourcing allocations, notify the Cost Service throughout the life-cycle of each process instance, passing the appropriate data for cost calculations. The workflow engine has also been extended to accommodate control-flow predicates that include cost-based expressions. When process execution reaches a control-flow predicate that contains a cost-based expression, the workflow engine calls the *Cost Service*, passing the expression, along with all associated

data. In addition, the set of resource allocation strategies within the YAWL Resource service has been extended with a number of cost-based allocators, such as *Cheapest Resource*, *Cheapest to Start*, *Cheapest Completer* and so on. When the Resource Service enacts a cost-based allocator at runtime, the allocator will directly query the *Cost Service*, requesting a calculation based on previous case histories (stored within the process logs) for the resources involved, based on the particular allocation strategy in question. Both push and pull based resource interaction styles are supported. With regards to process variants, the *Worklet Service* [9] will be extended with cost-based rule expressions, which may then be used to determine which process variant is the ideal selection for the current context of a case.

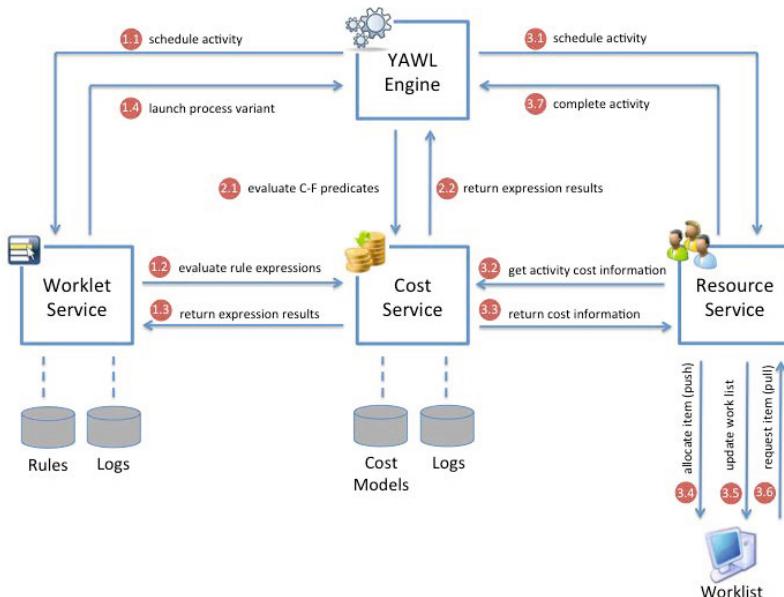


Fig. 3. Prototype architectural flow in the YAWL environment

Figure 3 shows the flow of information through the prototype for each level of cost-informed support. At the *process* level, the workflow engine schedules an activity for execution by the Worklet Service (1.1). The Worklet Service traverses its rule set for the activity, querying the Cost Service to evaluate cost-based rule expressions (1.2). The Cost Services evaluates and returns the results (1.3), which the Worklet Service uses to select the appropriate process variant for the activity, and launches the variant in the engine (1.4). At the *activity* level, when the workflow engine encounters an branching construct in the control-flow of a process instance, it queries the Cost Service to evaluate the predicate of each outgoing branch (2.1). The engine then uses the results of the predicate evaluations to fire the branch that evaluates to true (2.2). At the *resource* level, where the distribution of work takes place, the workflow engine schedules an activity for a (human) resource (3.1) with the Resource Service. The Resource

Service then queries the Cost Service for all cost information pertaining to the activity (3.2), which the Cost Service returns (3.3). If the activity is configured for system-based allocation (push pattern), the specified allocation strategy (e.g. Cheapest Resource) is employed using the cost information in its calculations, then the activity is routed to the worklist of the selected resource (3.4). If the activity is configured for resource-based allocation (pull pattern), the affected resources' worklists are updated with the retrieved cost information (3.5) allowing a resource to select the appropriate activity based on the cost information presented to them (3.6).

In addition to deploying process examples in the above prototype implementation realising our conceptual framework, we plan to evaluate the conceptual framework with stakeholders' input (e.g. through interviews and case studies).

4 Related Work

Cost has always been one of the key factors under consideration in the context of business process reengineering [5] and process improvements [7]. Through the iterative application of BPM techniques, processes can be improved in terms of quality, flexibility, time and/or cost [7]. Although WfMSs support planning, execution, (re)design and deployment of workflows [13], direct support for cost-informed execution is currently lacking. We have previously taken a first step by proposing a generic cost model [14], which is one of the ingredients of the encompassing framework we presented and demonstrated in the current paper.

The interrelationships between processes, resources and cost are also highlighted in the report produced by the International Federation of Accountants [6]. Notwithstanding these works, few studies exist where a structured approach to the analysis of cost factors in a process-aware information system is undertaken. Since the introduction of ERP systems, a number of studies have been conducted on the effects of ERP systems on traditional management accounting practices [1,3,4]. Recently, Vom Brocke et al. proposed an information model to link the ARIS accounting structure with ARIS process semantics using Event Driven Process Chains (EPC) [12]. Cost-informed operational process support is related to the notion of operational support studied in the context of process mining [10]. As shown in this paper, operational support based on cost considerations can be provided though an external cost service tightly coupled to the WfMS.

5 Conclusion and Future Work

The paper proposes a conceptual framework to enable workflow management systems to be cost-informed during enactment. In particular, we proposed how cost-based decision rules for process variant selections, activity related decisions (e.g., execution, cancellation, deferment), and resource assignment decisions can all be supported within a WfMS. We proposed an architecture for cost-informed process execution and presented a realisation of such a cost-informed workflow environment using the YAWL workflow management system.

We believe that our approach will enable organizations to more easily translate cost strategies into operational fulfilment using a WfMS and we have plans to evaluate the framework with stakeholders' input (e.g. through interviews and case studies). This work takes an important step towards achieving a higher level of support for WfMSs in terms of the cost perspective. For the future, we are interested in the development of predictive capabilities that may help to project the cost that is incurred by alternative operational decisions.

Acknowledgments. This work is supported by an ARC Discovery grant with number DP120101624.

References

1. Booth, P., Matolcsy, Z., Wieder, B.: The impacts of enterprise resource planning systems on accounting practice—the Australian experience. *Australian Accounting Review* 10(22), 4–18 (2000)
2. Chung, L., Nixon, B., Yu, E., Mylopoulos, J.: Non-functional requirements in software engineering. Kluwer (2000)
3. Grabski, S., Leech, S., Sangster, A.: Management accounting in enterprise resource planning systems. CIMA Publishing (2009)
4. Hyvönen, T.: Exploring Management Accounting Change in ERP Context. PhD thesis, University of Tampere (2010)
5. Kettinger, W., Teng, J., Guha, S.: Business process change: a study of methodologies, techniques, and tools. *MIS Quarterly* 21(1), 55–80 (1997)
6. Professional Accountants in Business Committee. Evaluating and improving costing in organizations (July 2009)
7. Reijers, H., Mansar, S.: Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega* 33(4), 283–306 (2005)
8. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow resource patterns: Identification, representation and tool support. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 216–232. Springer, Heidelberg (2005)
9. ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., Russell, N.: Modern Business Process Automation: YAWL and its support environment. Springer (2010)
10. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
11. van der Aalst, W.M.P., Dustdar, S.: Process mining put into context. *IEEE Internet Computing* 16, 82–86 (2012)
12. vom Brocke, J., Sonnenberg, C., Baumoel, U.: Linking Accounting and Process-Aware Information Systems - Towards a Generalized Information Model for Process-Oriented Accounting. In: European Conference on Information Systems, pp. 1–13 (2011)
13. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag, New York, Inc., Secaucus (2007)
14. Wynn, M.T., Low, W.Z., Nauta, W.: A framework for cost-aware process management: Generation of accurate and timely management accounting cost reports. In: Conferences in Research and Practice in Information Technology, CRPIT (2013)

Automating the Adaptation of Evolving Data-Intensive Ecosystems

Petros Manousis¹, Panos Vassiliadis¹, and George Papastefanatos²

¹ Dept. of Computer Science University of Ioannina (Hellas)

{pmanousi,pvassil}@cs.uoi.gr

² Research Center “Athena” (Hellas)

gpapas@imis.athenainnovation.gr

Abstract. Data-intensive ecosystems are conglomerations of data repositories surrounded by applications that depend on them for their operation. To support the graceful evolution of the ecosystem’s components we annotate them with policies for their response to evolutionary events. In this paper, we provide a method for the adaptation of ecosystems based on three algorithms that (i) assess the impact of a change, (ii) compute the need of different variants of an ecosystem’s components, depending on policy conflicts, and (iii) rewrite the modules to adapt to the change.

Keywords: Evolution, data-intensive ecosystems, adaptation.

1 Introduction

Data-intensive ecosystems are conglomerations of databases surrounded by applications that depend on them for their operation. Ecosystems differ from the typical information systems in the sense that the management of the database profoundly takes its surrounding applications into account. In this paper, we deal with the problem of facilitating the evolution of an ecosystem without impacting the smooth operation or the semantic consistency of its components.

Observe the ecosystem of Figure 1. On the left, we depict a small part of a university database with three relations and two views, one for the information around courses and another for the information concerning student transcripts. On the right, we isolate two queries that the developer has embedded in his applications, one concerning the statistics around the database course and the other reporting on the average grade of each student. If we were to delete attribute C_NAME, the ecosystem would be affected in two ways : (a) *syntactically*, as both the view V_TR and the query on the database course would crash, and, (b) *semantically*, as the latter query would no longer be able to work with the same selection condition on the course name. Similarly, if an attribute is added to a relation, we would like to inform dependent modules (views or queries) for the availability of this new information.

The means to facilitate the graceful evolution of the database without damaging the smooth operation of the ecosystem’s applications is to allow all the involved stakeholders to *register veto’s or preferences*: for example, we would

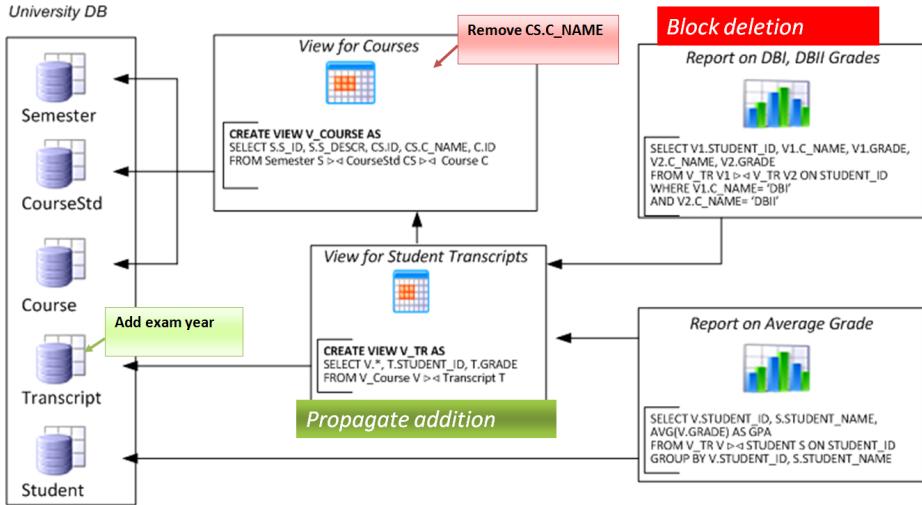


Fig. 1. Managing the adaptation of a University-DB Ecosystem

like to allow a developer to state that she is really adamant on retaining the structure and semantics of a certain view. In our method, we can annotate a *module* (i.e., relation, view or query) with a *policy* for each possible event that it can withstand, in one of two possible modes: (a) *block*, to veto the event and demand that the module retains its previous structure and semantics, or, (b) *propagate*, to allow the event and adapt the module to a new internal structure.

In this paper, we model ecosystems as graphs annotated with policies for responding to evolutionary events (Sec. 2) and we address the problem of identifying (a) what parts of the ecosystem are affected whenever we test a potential change and (b) how will the ecosystem look like once the implications of conflicting policies are resolved and the graph is appropriately rewritten (Sec. 3). Related work in ecosystem adaptation has provided us with techniques for view adaptation [1], [2], [3] that do not allow the definition of the policies for the adaptation of the ecosystem modules. Our previous work [4] has proposed algorithms for impact assessment with explicit policy annotation; however, to the best of our knowledge, there is no method that allows both the impact assessment and the rewriting of the ecosystem's modules along with correctness guarantees.

We implemented our method in a *what-if analysis* tool, Hecataeus¹ where all stakeholders can pre-assess the impact of possible modifications before actually performing them, in a way that is loosely coupled to the ecosystem's components. Our experimentation with ecosystems of different policies and sizes (Sec. 4) indicates that our method offers significant effort gains for the maintenance team of the ecosystem and, at the same time, scales gracefully.

¹ <http://www.cs.uoi.gr/~pvassil/projects/hecataeus/>

2 Formal Background

Our modeling technique, extending [4], uniformly represents all the components of an ecosystem as a directed graph which we call the *Architecture Graph* of the ecosystem. Fig. 2 visually represents the internals of the modules of Fig. 1. To avoid overcrowding the figure, we omit different parts of the structure in different modules; the figure is self-explanatory on this.

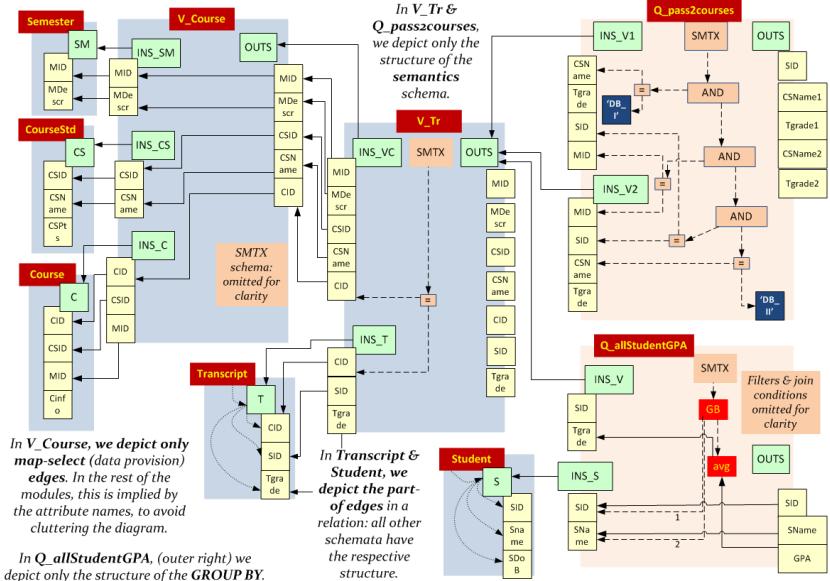


Fig. 2. A subset of the graph structure for the University-DB Ecosystem

Modules. A module is a semantically high level construct of the ecosystem; specifically, the modules of the ecosystem are relations, views and queries. Every module defines a scope recursively: every module has one or more schemata in its scope (defined by part-of edges), with each schema including components (e.g., the attributes of a schema or the nodes of a semantics tree) linked to the schema also via part-of edges. In our model, all modules have a well defined scope, “fenced” by input and output schemata.

Relations. Each relation includes a node for the relation per se, a node for its (output) schema and a node for each for its attributes; all connected via the aforementioned part-of edges.

Queries. The graph representation of a Select - Project - Join - Group By (SPJG) query involves a new node representing the query, named *query node*, linked to the following schemata:

1. a set of *input schemata nodes* (one for every table appearing in the FROM clause). Each input schema includes the set of attributes that participate in the syntax of the query (i.e., SELECT, WHERE and GROUP BY clauses, etc.). Each input attribute is linked via a provider, *map-select* edge to the appropriate attribute of the respective provider module.
2. an *output schema node* comprising the set of attributes present in the SELECT clause. The output attributes are linked to the appropriate input attributes that populate them through *map-select* edges, directing from the output towards the input attributes.
3. a *semantics node* as the root node for the sub-graph corresponding to the semantics of the query (specifically, the WHERE and GROUP-BY part).

We accommodate WHERE clauses in conjunctive normal form, where each atomic formula is expressed as: (i) Ωop constant, or (ii) $\Omega op \Omega'$, or (iii) $\Omega op Q$ where Ω, Ω' are attributes of the underlying relations, Q is a nested query, and operator op belongs to the set $\{<, >, =, \leq, \geq, \neq, IN, EXISTS, ANY\}$. The entire WHERE clause is mapped to a tree, where (i) each atomic formula is mapped to a subtree with an operator node for op linked with *operand* edges pointing to the operand nodes of the formulae and (ii) nodes for the Boolean operators (AND, OR) connect with each other as well as with the operators of the atomic formulae via the respective operand edges. The GROUP BY part is mapped in the graph via (i) a node GB, to capture the set of attributes acting as the aggregators and (ii) one node per aggregate function labeled with the name of the employed aggregate function; e.g., COUNT, SUM, MIN. For the aggregators, we use edges directing from the semantics node towards the GB node that are labeled *group-by*. The GB node is linked to the respective input attributes acting as aggregators with *group-by* edges, which are additionally tagged according to the order of the aggregators; we use an identifier i to represent the i -th aggregator. Moreover, for every aggregated attribute in the query's output schema, there exists a *map-select* edge directing from this attribute towards the aggregate function node as well as an edge from the function node towards the respective input attribute.

Views. Views are treated as queries; however the output schema of a view can be used as input by a subsequent view or query module.

Summary. A summary of the architecture graph is a zoomed-out variant of the graph at the schema level with provider edges only among schemata (instead of attributes too).

Events. We organize the events that can be tested via our method in the following groups.

- *Events at relations.* A relation can withstand deletion and renaming of itself as well as addition, deletion and renaming of its attributes.
- *Events at views and queries.* A view can withstand the deletion and renaming of itself, the addition, deletion or renaming of its output attributes and the update of the view's semantics (i.e., the modification of the WHERE clause of the respective SQL query that defines the view).

Policies. As already mentioned, the policy of a node for responding to an incoming event can be one of the following: (a) PROPAGATE, which means that the node is willing to adapt in order to be compatible with the new structure and semantics of the ecosystem, or, (b) BLOCK, which means that the node wants to retain the previous structure and semantics. We can *assign policies* to all the nodes of the ecosystem via a language [5] that provides guarantees for the complete coverage of *all* the graph's nodes along with syntax conciseness and customizability. The main idea is the usage of rules of the form <receiver node> : on <event> then <policy>, both at the default level –e.g.,

VIEW.OUT.SELF: on ADD_ATTRIBUTE then PROPAGATE;

and at the node-specific level (overriding defaults) –e.g.,

V_TR_OUT.SELF: on ADD_ATTRIBUTE then BLOCK;

3 Impact Assessment and Adaption of Ecosystems

The goal of our method is to assess the impact of a hypothetical event over an architecture graph annotated with policies and to adapt the graph to assume its new structure after the event has been propagated to all the affected modules. Before any event is tested, we topologically sort the modules of the architecture graph (always feasible as the summary graph is acyclic: relations have no cyclic dependencies and no query or view can have a cycle in their definition). This is performed once, in advance of any impact assessment. Then, in an on-line mode, we can perform what-if analysis for the impact of changes in two steps that involve: (i) the detection of the modules that are actually affected by the change and the identification of a status that characterizes their reaction to the event, and, (ii) the rewriting of the graph's modules to adapt to the applied change.

3.1 Detection of Affected Nodes and Status Determination

The assessment of the impact of an event to the ecosystem is a process that results in assigning every affected module with a status that characterizes its policy-driven response to the event. The task is reduced in (a) determining the affected modules in the correct order, and, (b) making them assume the appropriate status. Algorithm *Status Determination* (Fig. 3) details this process. In the following, we use the terms *node* and *module* interchangeably.

1. Whenever an event is assessed, we start from the module over which it is assessed and visit the rest of the nodes by following the topological sorting of the modules to ensure that a module is visited after *all* of its data providers have been visited. A visited node assesses the impact of the event internally (cf., "intra-module processing") and obtains a *status*, which can be one of the following: (a) BLOCK, meaning that the module is requesting that it remains structurally and semantically immune to the tested change and blocks the

Input: A topologically sorted architecture graph summary $\mathbf{G}_s(\mathbf{V}_s, \mathbf{E}_s)$, a global queue Q that facilitates the exchange of messages between modules.

Output: A list of modules $Affected\ Modules \subseteq \mathbf{V}_s$ that were affected by the event and acquire a status other than *NO_STATUS*.

1. $Q = \{\text{original message}\}$, $Affected\ Modules = \emptyset$;
2. For All $node \in \mathbf{G}_s(\mathbf{V}_s, \mathbf{E}_s)$
3. $node.status = NO_STATUS$;
4. EndFor
5. While $(size(Q) > 0)$
6. visit module ($node$) in head of Q ;
7. insert $node$ in *Affected Modules* list;
8. get all messages, *Messages*, that refer to $node$;
9. SetStatus($node$, *Messages*);
10. If ($node.status == PROPAGATE$) Then
11. insert $node$.*Consumers Messages* to the Q ;
12. EndWhile
13. Return *Affected Modules*;

Procedure SetStatus(*Module*, *Messages*)

Consumers Messages = \emptyset ;

For All *Message* \in *Messages*

 decide status of *Module*;

 put messages for *Module*'s consumers in *Consumers Messages*;

EndFor

Fig. 3. Algorithm STATUS DETERMINATION

event (as its immunity obscures the event from its data consumers), (b) PROPAGATE, meaning that the modules concedes to adapt to the change and propagate the event to any subsequent data consumers, or, (c) retain a NO_STATUS status, already assigned by the topological sort, meaning that the module is not affected by the change.

2. If the status of the module is PROPAGATE, the event must be propagated to the subsequent modules. To this end, the visited module prepares *messages* for its data consumers, notifying them about its own changes. These messages are pushed to a common *global message queue* (where messages are sorted by their target module's topological sorting identifier).
3. The process terminates whenever there are no more messages and no more modules to be visited.

Intra-module Processing. Whenever visited, a module starts by retrieving from the common queue *all* the messages (i.e., events) that concern it. It is possible that more than one message exist in the global queue for a module: e.g., with the deletion of an attribute that was used both in the output schema of a module and in the semantics schema of a module, the module should inform its consumers that (a) the attribute was deleted and (b) its semantics has changed. The processing of the messages is performed as follows:

1. First, the module probes its schemata for their reaction to the incoming event, starting from the input schemata, next to the semantics and finally to the output schema. Naturally, relations deal only with the output schema.
2. Within each schema, the schema has to probe both itself and its contained nodes (attributes) for their reaction to the incoming event. At the end of this process, the schema assumes a status as previously discussed.
3. Once all schemata have assumed status, it is the output schema of the module that decides the reaction of the overall module; if any of the schemata raises a veto (BLOCK) the module assumes the BLOCK status too; otherwise, it assumes the PROPAGATE status.

Theoretical Guarantees. Previous models of Architecture Graphs ([4]) allow queries and views to directly refer to the nodes representing the attributes of the involved relations. Due to the framing of modules within input and output schemata and the topological sorting, in [6] we have proved that the process (a) terminates and (b) correctly assigns statuses to modules.

3.2 Query and View Rewriting to Accommodate Change

Once the first step of the method, *Status Determination*, has been completed and each module has obtained a status, the problem of adaptation would intuitively seem simple: each module gets rewritten if the status is PROPAGATE and remains the same if the status is BLOCK. This would require only the execution of the *Graph Rewrite* step – in fact, one could envision cases where *Status Determination* and *Graph Rewrite* could be combined in a single pass. Unfortunately, although the decision on *Status Determination* can be made locally in each module, taking into consideration only the events generated by previous modules and the local policies, the decision on rewriting has to take extra information into consideration. This information is not local, and even worse, it pertains to the subsequent, consumer modules of an affected module, making thus impossible to weave this information in the first step of the method, *Status Determination*. The example of Fig. 4 is illustrative of this case.

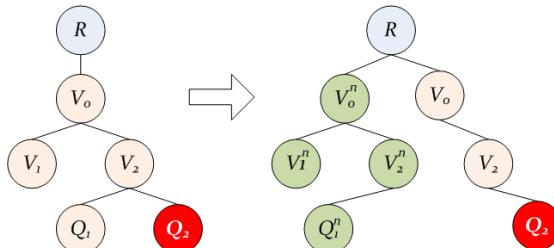


Fig. 4. Block rewriting example

In the example of Figure 4, we have a relation R and a view V_0 defined over the relation R . Two views (V_1 and V_2) use V_0 in order to get data. V_2 is further

used by two queries (Q_1 and Q_2). The database administrator wants to change V_0 , in a way that all modules depending on V_0 are going to be affected by that change (e.g., attribute deletion, for an attribute common to all the modules of the example). Assume now that all modules except Q_2 accept to adapt to the change, as they have a PROPAGATE policy annotation. Still, the vetoing Q_2 must be kept immune to the change; to achieve this we must retain the previous version of *all* the nodes in the path from the origin of the evolution (V_0) to the blocking Q_2 . As one can see in the figure, we now have two variants of V_0 and V_2 : the new ones (named V_0^n and V_2^n) that are adapted to the new structure of V_0 – now named V_0^n – and the old ones, that retain their name and are depicted in the rightmost part of the figure. The latter are immune to the change and their existence serves the purpose of correctly defining Q_2 .

Input: An architecture graph summary $\mathbf{G}_s(\mathbf{V}_s, \mathbf{E}_s)$, a list of modules *Affected modules*, affected by the event, and the *Initial Event* of the user.

Output: Annotation of the modules of *Affected modules* on the action needed to take, and specifically whether we have to make a new version of it, or, implement the change that the user asked on the current version

1. For All *Module* \in *Affected modules*
2. If(*Module.status* == *BLOCK*) Then
3. CheckModule(*Module*, *Affected modules*, *Initial Event*);
4. mark *Module* not to change; //Blockers do not change
5. EndFor

```
Procedure CheckModule(Module, Affected modules, Initial Event)
If(Module has been marked) Then return; //Notified by previous block path
If (Initial Event == ADD_ATTRIBUTE)
    Then mark Module to apply change on current version; //Blockers ignore provider addition
Else mark Module to keep current version as is and apply the change on a clone;
For All Module provider  $\in$  Affected modules feeding Module
    CheckModule(Module provider, Affected modules, Initial Event); //Notify path
EndFor
```

Fig. 5. Algorithm PATH CHECK

The crux of the problem is as follows: if a module has PROPAGATE status and none of its consumers (including both its immediate and its transitive consumers) raises a BLOCK veto, then both the module and all of these consumers are rewritten to a new version. However, if any of the immediate consumers, or any of the transitive consumers of a module raises a veto, then *the entire path towards this vetoing node must hold two versions of each module*: (a) the new version, as the module has accepted to adapt to the change by assuming a PROPAGATE status, and, (b) the old version in order to serve the correct definition of the vetoing module.

To correctly serve the above purpose, the adaptation process is split in two steps. The first of them, *Path Check*, works from the consumers towards the providers in order to determine the number of variants (old and new) for each

module. Whenever the algorithm visits a module, if its status is BLOCK, it starts a reverse traversal of the nodes, starting from the blocker module towards the module that initialized the flow and marks each module in that path (a) to keep its present form and (b) prepare for a cloned version (identical copy) where the rewriting will take place. The only exception to this rewriting is when the module of the initial message is a relation module and the event is an attribute deletion, in which case a BLOCK signifies a veto for the adaptation of the relation.

Input: A list of modules *Affected modules*, knowing the number of versions they have to retain, initial messages of *Affected modules*

Output: Architecture graph after the implementation of the change the user asked

1. If(any of *Affected modules* has status BLOCK) Then
2. If(initial message started from Relation module type AND event == DELETE_ATTRIBUTE) Then Return;
3. Else
4. For All (*Module* ∈ *Affected modules*)
5. If(*Module* needs only new version) Then
6. proceed with rewriting of *Module*;
7. connect *Module* to new providers; //new version goes to new path
8. Else
9. clone *Module*; //clone module, to keep both versions
10. connect cloned *Module* to new providers; //clone is the new version
11. proceed with rewriting of cloned *Module*;
12. EndFor
13. Else
14. For All *Module* ∈ *Affected modules*
15. proceed with rewriting of *Module* //no blocker node
16. EndFor

Fig. 6. Algorithm GRAPH REWRITE

Finally, all nodes that have to be rewritten are getting their new definition according to their incoming events. Unfortunately, this step cannot be blended with *Path Check* straightforwardly: *Path Check* operates from the end of the graph backwards, to highlight cases of multiple variants; rewriting however, has to work from the beginning towards the end of the graph in order to correctly propagate information concerning the rewrite (e.g., the names of affected attributes, new semantics, etc.). So, the final part of the method, *Graph Rewrite*, visits each module and rewrites the module as follows:

- If the module must retain only the new version, once we have performed the needed change, we connect it correctly to the providers it should have.
- If the module needs both the old and the new versions, we make a clone of the module to our graph, perform the needed change over the cloned module and connect it correctly to the providers it should have.
- If the module retains only the old version, we do not perform any change.

4 Experiments

We assessed our method for its usefulness and scalability with varying graph configurations and policies; in this section, we report our findings.

Experimental Setup. We have employed TPC-DS, version 1.1.0 [7] as our experimental testbed. TPC-DS is a benchmark that involves star schemata of a company that has the ability to *Sell* and receive *Returns* of its *Items* with the following ways: (a) the *Web*, or, (b) a *Catalog*, or, (c) directly at the *Store*. Since the Hecataeus' parser could not support all the advanced SQL constructs of TPC-DS, we employed several auxiliary views and slight query modifications.

Graphs and Events. To test the effect of graph size to our method's efficiency, we have created 3 graphs with gradually decreasing number of query modules: (a) a large ecosystem, *WCS*, with queries using all the available fact tables, (b) an ecosystem *CS*, where the queries to WEB_SALES have been removed, and (c) an ecosystem *S*, with queries using only the STORE_SALES fact table. The event workload consists of 51 events simulating a real case study of the Greek public sector. See Fig. 7 for an analysis of the module sizes within each scenario and the workload (listing the percentage of each event type as *pct*).

Policies. We have annotated the graphs with policies, in order to allow the management of evolution events. We have used two “profiles”: (a) *MixtureDBA*, consisting of 20% of the relation modules annotated with BLOCK policy and (b) *MixtureAD*, consisting of 15% of the query modules annotated with BLOCK policy. The first profile corresponds to a developer-friendly DBA that agrees to prevent changes already within the database. The second profile tests an environment where the application developer is allowed to register veto's for the evolution of specific applications (here: specific queries). We have taken care to pick queries that span several relations of the database.

	Graph size			Event type	pct
	S	CS	WCS		
Queries	27	68	89	Attribute Add	37.3%
Views	25	48	95	Attribute Rename	43.2%
Relations	25	25	25	Attribute Del	13.7%
Sum	77	141	218	Relation Rename	1.9%
				View alter semantics	3.9%

Fig. 7. Experimental configuration for the TPC-DS ecosystem

Experimental Protocol. We have used the following sequence of actions. First, we annotate the architecture graph with policies. Next, we sequentially apply the events over the graph – i.e., each event is applied over the graph that resulted from the application of the previous event. We have performed our experiments with hot cache. For each event we measure the elapsed time for each of the three algorithms, along with the number of affected, cloned and adapted modules. All the experiments have been performed in a typical PC with an Intel Quad core CPU at 2.66GHz and 1.9GB main memory.

Effectiveness. How useful is our method for the application developers and the DBA's? We can assess the effort gain of a developer using the highlighting of affected modules of Hecataeus compared to the situation where he would have to perform all checks by hand as the *fraction of Affected Modules of the ecosystem*. This gain, expressed via the $\%AM$ metric amounts to the percentage of useless checks the user would have made. We exclude the object that initiates the sequence of events from the computation, as it would be counted in both occasions. Formally, $\%AM$ is given by the Equation 1.

$$\%AM = 1 - \frac{\# \text{Affected Modules}}{\#(\text{Queries} \cup \text{Views})} \quad (1)$$

	$\%AM - \text{Mixture AD}$			$\%AM - \text{Mixture DBA}$		
	S	CS	WCS	S	CS	WCS
min	21%	35%	30%	60%	78%	84%
avg	89%	91%	92%	97%	96%	97%
max	100%	100%	100%	100%	100%	100%

Fig. 8. Effectiveness assessment as fraction of affected modules ($\%AM$)

The results depicted in Fig. 8 demonstrate that the effort gains compared to the absence of our method are significant, as, on average, the effort is around 90% in the case of the AD mixture and 97% in the case of the DBA mixture. As the graph size increases, the benefits from the highlighting of affected modules that our method offers, increase too. Observe that in the case of the DBA case, where the flooding of events is restricted early enough at the database's relations, the minimum benefit in all 51 events ranges between 60% - 84%.

Effect of Policy to the Execution Time. In the case of *Mixture DBA* we follow an aggressive blocking policy that stops the events early enough, at the relations, before they start being propagated in the ecosystem. On the other hand, in the case of *Mixture AD*, we follow a more conservative annotation approach, where the developer can assign blocker policies only to some module parts that he authors. In the latter case, it is clear that the events are propagated to larger parts of the ecosystem resulting in higher numbers of affected and rewritten nodes. If one compares the execution time of the three cases of the AD mixture in Fig. 9 with the execution time of the three cases of the DBA mixture the difference is in the area of one order of magnitude. It is however interesting to note the internal differences: the status determination time is scaled up with a factor of two; the rewriting time, however is scaled up by a factor of 10, 20 and 30 for the small, medium and large graph respectively!

Another interesting finding concerns the **internal breakdown of the execution time** in each case. A common pattern is that *path check is executed very efficiently*: in all cases it stays within 2% of the total time (thus practically invisible in the graphic). In the case of the AD mixture, the analogy between the status determination and the graph rewriting part starts from a 24% - 74% for

the small graph and ends to a 7% - 93% for the large graph. In other words, *as the events are allowed to flow within the ecosystem, the amount of rewriting increases with the size of the graph*; in all cases, it dominates the overall execution time. This is due to the fact that rewriting involves memory management (module cloning, internal node additions, etc) that costs much more than the simple checks performed by *Status Determination*. In the case of the DBA mixture, however, where the events are quickly blocked, the times are not only significantly smaller, but also equi-balanced: 57% - 42% for the small graph (*Status Determination* costs more in this case) and 49% - 50% for the two other graphs. Again, this is due to the fact that the rewriting actions are the time consuming ones and therefore, their reduction significantly reduces the execution time too.

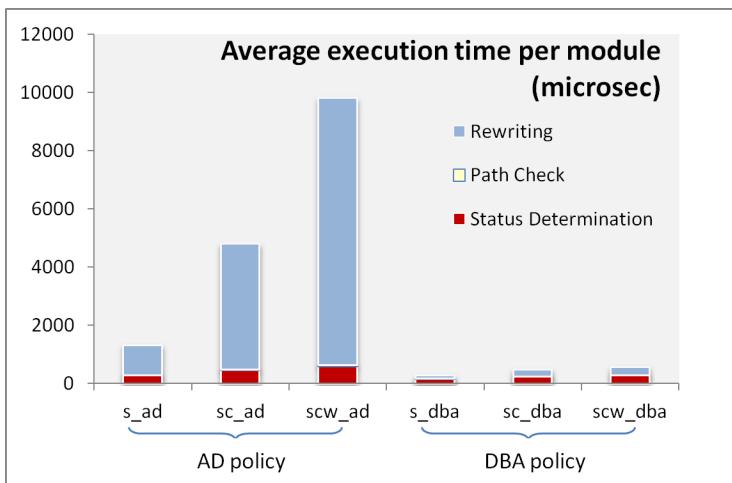


Fig. 9. Efficiency assessment for different policies, graph sizes and phases

Effect of Graph Size to the Execution Time. To assess the impact of graph size to the execution time one has to compare the three different graphs to one another within each policy. In the case of the AD mixture, where the rewriting dominates the execution time, there is a linear increase of both the rewriting and the execution time with the graph size. On the contrary, the rate of increase drops in the case of the DBA mixture: when the events are blocked early, the size of the graph plays less role to the execution time.

Overall, the main lesson learned from these observations is that the annotation of few database relations significantly restricts the rewriting time (and consequently the overall execution time) when compared to the case of annotating modules external to the database. In case the rewriting is not constrained early enough, then the execution cost grows linearly with the size of the ecosystem.

5 Related Work

For an overview of the vast amount of work in the area of evolution, we refer the interested reader to an excellent, recent survey [8]. We also refer the interested reader to [9] for a survey of efforts towards bidirectional transformations. Here, we scope our discussion to works that pertain to the adaptation of data-intensive ecosystems.

Data-Intensive Ecosystems’ Evolution. Research activity around data-intensive ecosystems has been developed around two tools, Hecataeus and Prism. Hecataeus [4] models ecosystems as Architecture Graphs and allows the definition of policies, the impact assessment of potential changes and the computation of graph-theoretic properties as metrics for the vulnerability of the graph’s nodes to change. The impact assessment mechanism was first introduced in [4] and subsequently modified in [6]. PRISM++ [10] lets the user define his policies about imminent changes. The authors use ICMOs (Integrity Constraints Modification Operators) and SMOs (Schema Modification Operators) in order to rewrite the queries/views in a way that the results of the query/view are the same as before.

View/Schema Mapping Rewriting. Nica et al., [1] make legal rewritings of views affected by changes and they primarily deal with the case of relation deletion by finding valid replacements for the affected (deleted) components via a meta-knowledge base (MKB) that keeps meta-information about attributes and their join equivalence attributes on other tables in the form of a hyper-graph. Gupta et al., [2] redefine a *materialized* view as a sequence of primitive local changes in the view definition. On more complex adaptations, those local changes can be pipelined in order to compute the new view contents incrementally and avoid their full re-computation. Velegrakis, et al., [3], deal with the maintenance of a set of mappings in an environment where source and target schemata are integrated under schema mappings implemented via SPJ queries. Cleve et al., [11] introduce mappings among the applications and a conceptual representation of the database, again mapped to the database logical model; when the database changes, the mappings allow to highlight impacted areas in the source code.

Comparison to Existing Approaches. As already mentioned, the annotation of the ecosystem with policies imposes the new problem of maintaining different replicas of view definitions for different consumers; to the best of our knowledge, this is the first time that this problem is handled in a systematic way. Interestingly, although the existing approaches make no explicit treatment of policies, they differ in the implicit assumptions they make. Nica et al., operating mainly over virtual views [1], actually block the flooding of a deletion event by trying to compensate the deletion with equivalent expressions. At the same time, they do not handle additions or renamings. Velegrakis et al. [3] move towards the same goal but only for SPJ queries. On the other hand, Gupta et al., [2], working with materialized views, are focused to adapting the contents of the views, in a propagate-all fashion. A problem coming with a propagate-all policy is that the events might affect the semantical part of the views/queries (WHERE clause)

without any notification to the involved users (observe that the problem scales up with multiple layers of views defined over other views).

Compared to previous editions of Hecataeus [4], this work reports on the first implementation of a status determination mechanism with correctness guarantees. The management of rewritings via the path checking to handle conflicting policies and the adaptation to accommodate change are completely novel.

6 Conclusions and Future Work

In this paper we have addressed the problem of adapting a data-intensive ecosystem in the presence of policies that regulate the flow of evolution events. Our method allows (a) the management of alternative variants of views and queries and (b) the rewriting of the ecosystem's affected modules in a way that respects the policy annotations and the correctness of the rewriting (even in the presence of policy conflicts). Our experiments confirm that the adaptation is performed efficiently as the size and complexity of the ecosystem grow. Future work can address the assessment of complicated events, the visualization of the ecosystem and the automatic suggestion of policies.

Acknowledgments. This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

References

1. Nica, A., Lee, A.J., Rundensteiner, E.A.: The CVS Algorithm for View Synchronization in Evolvable Large-Scale Information Systems. In: Schek, H.-J., Salton, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, pp. 359–373. Springer, Heidelberg (1998)
2. Gupta, A., Mumick, I.S., Rao, J., Ross, K.A.: Adapting materialized views after redefinitions: techniques and a performance study. *Information Systems* 26(5), 323–362 (2001)
3. Velegrakis, Y., Miller, R.J., Popa, L.: Preserving mapping consistency under schema changes. *VLDB Journal* 13(3), 274–293 (2004)
4. Papastefanatos, G., Vassiliadis, P., Simitsis, A., Vassiliou, Y.: Policy-Regulated Management of ETL Evolution. *J. Data Semantics* 13, 147–177 (2009)
5. Manousis, P.: Database evolution and maintenance of dependent applications via query rewriting. MSc. Thesis, Dept. of Computer Science, Univ. Ioannina (2013), <http://www.cs.uoi.gr/~pmanousi/publications.html>
6. Papastefanatos, G., Vassiliadis, P., Simitsis, A.: Propagating evolution events in data-centric software artifacts. In: ICDE Workshops, pp. 162–167 (2011)
7. Transaction Processing Performance Council: The New Decision Support Benchmark Standard (2012), <http://www.tpc.org/tpcds/default.asp>

8. Hartung, M., Terwilliger, J.F., Rahm, E.: Recent Advances in Schema and Ontology Evolution. In: Schema Matching and Mapping, pp. 149–190. Springer (2011)
9. Terwilliger, J.F., Cleve, A., Curino, C.: How clean is your sandbox? - towards a unified theoretical framework for incremental bidirectional transformations. In: 5th Intl. Conf. Theory and Practice of Model Transformations (ICMT), Prague, Czech Rep., pp. 1–23 (2012)
10. Curino, C., Moon, H.J., Deutsch, A., Zaniolo, C.: Update Rewriting and Integrity Constraint Maintenance in a Schema Evolution Support System: PRISM++. PVLDB 4(2), 117–128 (2010)
11. Cleve, A., Brogneaux, A.-F., Hainaut, J.-L.: A conceptual approach to database applications evolution. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 132–145. Springer, Heidelberg (2010)

sonSchema: A Conceptual Schema for Social Networks*

Zhifeng Bao, Y.C. Tay, and Jingbo Zhou

National University of Singapore

Abstract. sonSQL is a MySQL variant that aims to be the default database system for social network data. It uses a conceptual schema called sonSchema to translate a social network design into logical tables. This paper introduces sonSchema, shows how it can be instantiated, and illustrates social network analysis for sonSchema datasets. Experiments show such SQL-based analysis brings insight into community evolution, cluster discovery and action propagation.

Keywords: schema, social network.

1 Introduction

The proliferation of online social networks is now as unstoppable as the spread of the Web. Such a network needs a database system to properly manage its data. Many social network services are started by small teams of developers and typically use some free, off-the-shelf database system, e.g. MySQL. If the service is successful and the team grows to include professional database administrators, the dataset may be so large already that any re-engineering of the early decisions becomes difficult.

Our contribution to solving this problem is **sonSQL** (<http://sonsql.comp.nus.edu.sg>), a variant of MySQL that we hope to develop into the default database management system for social networks. We provide sonSQL with an interactive user interface and a rule-based expert system to translate a developer's design for a social network into a set of db-relational¹ tables. These tables are logical instantiations of a conceptual schema, called **sonSchema** ("son" for "social network"), that we designed for social network data. The objective of this paper is to introduce sonSchema.

We follow two guidelines in our design of sonSchema: **(G1: Generality)** The schema should be sufficiently general that it can model any social network design from the developer; **(G2: Service-Oriented)** The entities and relationships in the schema must correspond naturally to social network activity and services.

Given our goal of building a database system for social networks, why chose a db-relational system (MySQL) as the code base? There is a trend in the convergence of OLTP and OLAP², so social network analysis (SNA) may increasingly run on live data.

* This research was supported in part by MOE Grant No. R-252-000-394-112 and carried out at the SeSaMe Centre, which is supported by the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO.

¹ Following Angles and Gutiérrez [2], we use "db-relation" to refer to a database table, while "sn-relation" refers to a relationship in a social network.

² <http://www.greenplum.com/products/greenplum-uap>

Can a sonSchema dataset support SNA effectively? Most SNA in the literature are based on graph models, so should we start with a graph database system instead?

We studied one well-known problem in SNA, namely link prediction [15]. We developed **sonLP**, a predictor that applies principal component regression to features from multiple dimensions in the data [5]. For the example of a social graph defined by coauthorship among ACM authors, the links are in one dimension, while affiliation, research areas, etc., are in other dimensions. Experiments on such data show that sonLP outperforms HPLP+, a state-of-the-art predictor that is based entirely on the social graph [16]. This suggests that SNA should work with features *beyond* the graph topology.

By itself, sonLP cannot make the case for choosing db-relations instead of graphs as the data model. A fuller discussion of this choice is in our technical report [4]. Perhaps the most compelling argument, for us, is this: A database system for social network data must have an expressive query language, query optimization, indices, integrity constraints, concurrency control, crash recovery, batch processing and data exploration tools. Implementing this list to bring a prototype to market is highly non-trivial for any database system, and the only ones to have done so are db-relational. This may be why the core databases for Facebook (<http://www.facebook.com>) and Twitter (<http://www.twitter.com>) remain db-relational, even though some of their data use NoSQL (e.g. Cassandra for Twitter) [18].

The NoSQL movement (nosql-database.org) has argued that db-relational systems do not suit massive datasets because ACID consistency would compromise partition tolerance. However, there are other failures besides network partitioning and, again, the task of implementing some non-ACID consistency to handle various failures can be overwhelming [21]; this is particularly true for start-ups that need off-the-shelf systems.

We therefore chose to go with a db-relational system. It remains for us to show (later in this paper) that SQL-based SNA can even provide better insight (in some perspective) than analyzing some graph extracted from the data.

This paper makes three contributions: (1) We introduce sonSchema, a conceptual schema for social network data. (2) We show how social network analysis can be done with SQL queries on a sonSchema dataset. (3) We present insights on community evolution, cluster discovery and action propagation. Such insights are hard to extract from just the social graph because they require multi-dimensional access to the raw data, using the full range of db-relational query operators.

We begin by introducing sonSchema in Sec. 2, and present several examples of instantiations of sonSchema in Sec. 3. Sec. 4 then demonstrates how three well-known problems in social network analysis can be studied for a sonSchema dataset. Results from experiments, reported in Sec. 5, show that these techniques are effective and provide new insights for these problems. Sec. 6 reviews related work, before Sec. 7 concludes with a brief description of current work.

2 From Guidelines to Conceptual and Logical Schemas

Sec. 2.1 first characterizes a social network. The guidelines (G1) and (G2) then lead us to the conceptual schema. The db-relational form of this schema is sonSchema, which we present in Sec. 2.2, together with examples of translation into logical schemas.

2.1 Social Network Entities and Relationships

For generality (G1), we start with the following fundamental characterization [25]:

An online social network is a group of users who interact through social products.

This informal definition focuses on social interaction, and explicitly points out the role of products (games, events, songs, polls, etc.). It suggests four entities for our model:

- (E1) **user** is generic; it can be Jim, an advertiser, a retailer, etc.
- (E2) **group** has details (names, membership size, etc.) of an interest group.
- (E3) **post** may be a blog, tag, video, etc. contributed by a user; it includes the original post, comments on that post, comments on those comments, etc.
- (E4) **social_product** is a product with some intended consumers, like an event created for a group, a retailer's coupon for specific customers, etc. A **post** can be considered as a special case of a **social_product** (that has no intended consumer).

Similarly, there are four natural relationships:

- (R1) **friendship** among users may refer to Twitter followers, ACM coauthors, etc.
- (R2) **membership** connects a user to a group.
- (R3) **social_product_activity** connects a user to a social product through an activity (buy a coupon, vote in a poll, etc.).
- (R4) **social_product_relationship** connects two social products, like between a meeting and a poll, or between a charity event and a sponsor's discount coupons.

Interaction creates another entity and relationship:

- (E5) **private_msg** is a message that is visible only to the sender and receiver(s).
- (R5) **response2post** is a relationship between a tag and an image, a comment and a blog, a comment and another comment, etc. If **post** is considered a **social_product**, then one can consider **response2post** as a special case of **social_product_relationship**.

The above exhausts the list of entities (users and products) and relationships (user-user, user-product, product-product) in any online social network, in line with the generality requirement (G1). The refinements of **social_product** into **post** and **private_msg** reflect services that are usually provided by social networks (G2). Service orientation (G2) also guides our model for the interactions that give life to a social network; e.g. we split a comment into a **post** ((E3) modeling the content) and a **response2post** ((R5) modeling the interaction).

2.2 From Conceptual to Logical Schemas

sonSchema is the db-relational form, shown in Fig. 1, of the conceptual schema in Sec. 2.1. In the following, we use **bold** for a schema, *italics* for an instance of the schema, and typewriter font for attributes.

The table **friendship** stores user pairs, since most db-relational systems do not provide an attribute type suitable for lists of friends. Fig. 1 shows an optional attribute **group_id** in **post** for the case where a post (e.g. a paper) belongs to a group (e.g. journal). There is another optional attribute, **product_group**, for the category that this

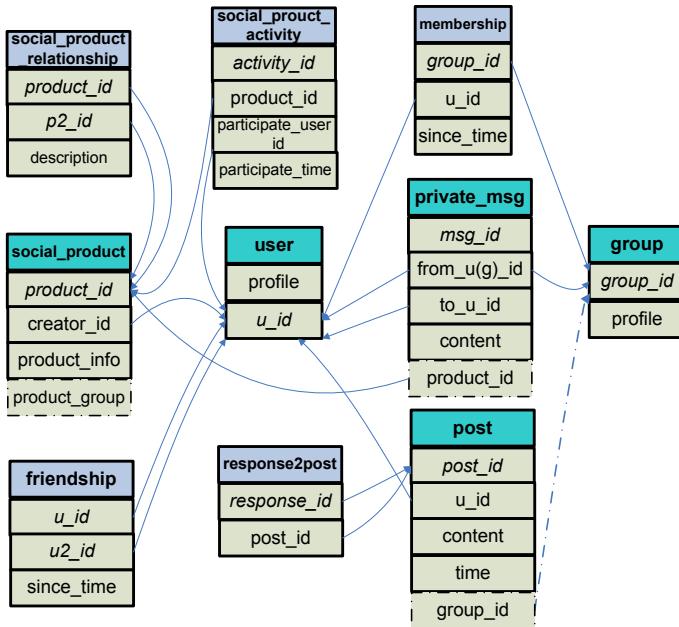


Fig. 1. sonSchema. Table names are in **bold**, primary keys are in *italics*, and edges point from foreign key to primary key. The 5 green tables are for entities, the 5 blue tables for relationships. Some attributes (e.g. *group_id* in **post**) are optional.

product belongs to. E.g. a social network for sharing and recommendation, like Douban (<http://www.douban.com>), may categorize products into books (further classified by topics like history, fiction, etc), movies, etc.

Fig. 1 may appear to be a logical schema, with a table each for the 5 entities and 5 relationships in Sec. 2.1. Rather, like the snowflake schema [14], sonSchema is in fact a conceptual schema that can be translated into different logical schemas.

For example, **private_msg** can be instantiated as two tables, one for messages between two users, and one from a group to its members (hence the attribute *from_u(g)_id* in Fig. 1). Similarly, **social_product** in Fig. 1 has a *creator_id* that may be a *u_id* or a *group_id*. Thus, Facebook events (<http://socialmediadiyworkshop.com/2010/03/manage-your-facebook-event/>), like a wedding organized by a bride or a rally organized by a union, can belong to different tables. Also, to allow a **post** to be shared (e.g. retweeted) within a restricted group, one can add a *shared_to* table, as an additional instance of **social_product_activity**.

Adding new attributes is costly for a db-relational system. To overcome this problem, we make sonSchema extensible through multiple instantiations of its table schemas. A social network service provider that wants to introduce a new service (image repository, event organizer, etc.) just adds corresponding tables for **social_product**, **post**, etc.

This again illustrates how the sonSchema design is service-oriented (G2), and its extensibility provides generality (G1).

3 Application: Social Network Data Management

We now verify that sonSchema can be instantiated for some social network services, and consider two applications: the ACM Digital Library (we will later use it to analyze the social network of ACM authors) and coupon dissemination via a social network.

3.1 Online Social Networks

Online social networks are driven by an entity (image, video, etc.), thus generating a corresponding set of tables. E.g. when Jay uploads a photograph (a **post**), Kaye may comment on it and Elle may tag Kaye (two separate instantiations of **response2post**), while mum may email Jay about the photograph (a **private_msg**). We have designed sonSchema to match this general structure of social interactions, so we expect its restricted form to suffice for current and future social networks.

The following considers the instantiation in greater detail:

Undirected Social Graphs. Facebook, Renren and LinkedIn are major social networks, where **friendship** models classmates, colleagues, etc. **post** or **social_product** models “status”, “note”, “photo”, “event”, etc., and **response2post** or **social_product_activity** models “like”, “comment”, “share”, “join_event”, “tag”, “via”, etc.

Directed Social Graphs. Twitter and Sina Weibo (<http://www.weibo.com>) are large social networks with directed follower-followed sn-relationships. Again, **post** would model tweets, and **response2post** would model activities like “retweet”, “comment” and “reply”.

Mixed Social Graphs. For a service like Flickr, **friendship** can have two instantiations, one for directed and the other for undirected links in user contact lists.

3.2 ACM Digital Library (ACMDL)

Facebook users often belong to overlapping social networks, some implicitly defined. Can sonSchema model such networks? Since Facebook data is not publicly available, we use ACMDL as a proxy: the publications therein define at least two social networks — an explicit bidirectional coauthorship, and an implicit directional citation.

In detail [4], sonSchema can model ACMDL with **user** instantiated as *author*, **post** as *papers* (with **group_id** identifying the publication venue), **friendship** as *coauthorship*, **response2post** as *citation*, and **group** as *conference/journal*. Many implicit social networks can be found via selection (e.g. **affiliation**) and joins of these tables.

3.3 Coupon Dissemination

To extract value from a social network, a service provider like Groupon (<http://www.groupon.com>) may use it to disseminate coupons for businesses. sonSchema can model such a service by instantiating **user** as *business* and *consumer* (2 tables), **social_product** as *coupon*, and **private_msg** as *coupon_dissemination*; coupon forwarding among members of a social group [19] is a separate instance of **private_msg**.

4 Application: Social Network Analysis

We want to verify that db-relations are better than graphs for modeling and analyzing social network data. To do so, we examine three well-studied problems in the literature, namely community structure (Sec. 4.1), cluster discovery (Sec. 4.2) and action propagation (Sec. 4.3). In particular, we show how such social network analysis can be done with SQL queries, instead of graph algorithms.

4.1 Community Structure

Current techniques for studying community definition and evolution invariably use graphs. We now examine how this can be done differently, using sonSchema.

Community Definition. Some social networks are explicitly declared [3] (e.g. LiveJournal, <http://www.livejournal.com>), but many are only implicitly defined. With sonSchema, one can easily discover, say, camera fans or bird watchers in the Flickr data by querying **response2post** for relevant tags. Similarly, one can extract the communities for data mining and cloud computing from ACMDL by specifying relevant journals and conferences, or keywords in the attribute **abstract** for *papers*.

In social network analysis, the interaction graph is most important [25]. Since interaction is explicitly represented by **private_msg**, **response2post** and **social_product_activity**, extracting the interaction graph is easy with sonSchema. In the Flickr example, if one defines an interaction as two users commenting on each other's photographs, then such user pairs can be retrieved with an appropriate join query.

Community Growth. One way of studying the growth of a community C is to determine the probability that someone on the fringe of C joins C [3]. Let $\text{fringe}(C)$ consist of users who are not in C but have a friend in C . For integer $K \geq 0$, let $\text{prob}(C|K, \Delta T)$ be the probability that a user $x \in \text{fringe}(C)$ will join C within the next time period ΔT if x has K friends in C .

We can compute $\text{prob}(C|K, \Delta T)$ in three steps: (i) run queries to take two snapshots of C — C_1 at time T_1 and C_2 at time $T_2 = T_1 + \Delta T$; (ii) retrieve those x in $\text{fringe}(C_1)$ with K friends in C_1 ; (iii) count how many x from (ii) are in C_2 .

Note that, in the above, there are other natural variations in the definition of $\text{fringe}(C)$ and $\text{prob}(C|K, \Delta T)$, as illustrated later in Sec. 5.1.

4.2 Cluster Discovery

A social network often contains clusters, whose members interact more among themselves than with others outside their cluster. E.g. Facebook users may have clusters from the same school or club. We now illustrate cluster discovery via sonSchema.

For better clarity, we use the sonSchema model of ACMDL from Sec. 3.2. (Here, a cluster may form around Codd, or Knuth.) Let coauthorship frequency of an author x be the sum of number of coauthors on all papers written by x (if coauthor y appears on n papers, y is counted n times).

Algorithm 1. Cluster discovery for a social network of authors

```

input : sonSchema dataset for publications,  $K$ : choice of top- $K$  clusters,  $\tau$ : threshold for
      a qualified coauthorship frequency
output: a set of clusters:  $CSet$ 
1 for each  $au \in author$  do
    /* coauthor_freq(author_id, freq, isVisited) is a table */ 
2   coauthor_freq( $au$ ).freq = number of coauthorships per author  $au$  (SQL1);
3 repeat
4   let  $a = au$  that has the highest  $coauthor\_freq.freq$  and not isVisited (SQL2);
5   let cluster  $C = \{a\}$  initially;
6   let queue  $Q = \{a\}$  initially;
7   let  $P = \emptyset$  initially;
8   repeat
9     let  $a = Q.pop()$ ;
10    let  $B = \{b \mid (\langle a, b \rangle \in \text{coauthor} \text{ OR } \langle b, a \rangle \in \text{coauthor}) \text{ AND } (!b.\text{isVisited})\}$  (SQL3);
11    for each  $b \in B$  do
12      let  $P_{a,b} = \text{set of paper_ids coauthored by } \langle a, b \rangle$ ;
13      if ( $coauthor\_freq(b,C) > \tau$ ) AND ( $\exists p \in P_{a,b} \text{ such that } p \notin P$ ) (SQL4) then
14         $C.add(b)$ ;
15         $Q.push(b)$ ;
16         $P.add(p)$ ;
17         $coauthor.freq(b).\text{isVisited} = \text{true}$ ;
18        /* mark  $b$  as visited in table author_freq */ 
19    until ( $Q$  is empty);
20   Output cluster  $C$ ;
21 until (The K-th result has been found);

```

Our Algo. 1 (above) for cluster discovery returns the top- K clusters, defined by using an appropriate quality metric, like coauthorship frequency and number of papers. It has an outer loop to find K clusters, and an inner loop to grow each cluster. It finds the coauthorship frequency of all authors (**SQL1**), then picks the author with the highest frequency to grow a cluster C (**SQL2**). Because a qualified community will most likely include these active authors. The latter is done by iteratively picking from coauthors of those in C (**SQL3**) and applying the quality metric (**SQL4**). This metric has a threshold τ for coauthorship frequency between a pair of authors.

A reasonable choice is $\tau = f/n$, where f is the total number of coauthorships and n is the number of authors in the cluster, so $coauthor.freq(b,C) > f/n$ (**SQL4**) means that adding b to C would raise the average coauthorship in the cluster. Thus, only authors with strong coauthorship ties with C are admitted.

The other quality metric (**SQL4**) requires that b cannot join C if b cannot add a new paper for C . Intuitively, a cluster with more papers is better.

Algo. 1 stops trying to grow C when it cannot be expanded (line18). It then picks another author as seed to grow the next cluster, unless there are already K clusters.

The crucial parts of Algo. 1 are straightforward SQL select-join queries, and critical calculations concerning quality can be easily coded and efficiently executed. There is no use of graph algorithms like network flow [8], spectral algorithms [1] and hierarchical decomposition [6]. Clustering quality is judged semantically by coauthorships per

author and number of papers per cluster, rather than syntactically by conductance [10] and modularity [6], etc. One can use a small K to generate only the most significant clusters, without having to decompose the entire graph. Similarly, one does not need to load the whole graph into memory (a problem for massive datasets); rather, Algo. 1's memory requirement is of the order of the cluster size, which is usually small [13].

4.3 Action Propagation

Social network analysis should focus on user interaction, as that is the very reason people join these networks. In current efforts to extract value from social networks (e.g. coupon dissemination in Sec. 3.3), one key idea is that friends influence one another.

Analyzing action propagation through a network starts with a log of user actions [9]. With sonSchema, we can extract this log as a table $Action(u, \alpha, t_u)$ from, say, **social_product_activity**, indicating user u performed action α at time t_u . We say α propagates from u to v if and only if $\langle u, v \rangle$ is an edge, $\langle u, \alpha, t_u \rangle \in Action$ and $\langle v, \alpha, t_v \rangle \in Action$ for some $t_u < t_v$, and $\langle u, v \rangle$ formed before t_v .

This definition for action propagation can be easily evaluated with a SQL query, and Sec. 5.3 will demonstrate this with a multi-dimensional analysis of Twitter dynamics.

5 Experiments

We now present experiments with the algorithms described in Sec. 4. The experiments are done with one dataset from the ACM Digital Library³ and another from Twitter (publicly available). Our ACMDL dataset has 445656 authors, 265381 papers, 4291 journals and conferences, 1741476 citation pairs and 170020 coauthorships. The schemas for these datasets are described in Sec. 3.1 and Sec. 3.2. All experiments are run on a Linux machine with 1.15GHz AMD Opteron(tm) 64-bit processor and MySQL (version 5.1.51). No index is built on non-key attributes.

Before experimenting with our SQL-based algorithms for social network analysis, we should do a head-to-head comparison between MySQL and a graph database system. Where a social network is modeled as a graph and for queries that are expressible as graph traversal, one expects a graph database system will have an advantage over a db-relational system in processing the queries. This is because graph traversal corresponds to table joins, which is an expensive operation. However, graph traversal in db-relational systems can be accelerated via the use of an appropriate index.

For the comparison, we use Neo4j (a leading graph database system) and equip MySQL with a GRIPP index [23] that is implemented as stored procedures to support efficient graph traversal. We interpreted *coauthorship* in our ACMDL dataset as an undirected graph, with 59695 nodes (authors) and 106617 edges (coauthorships).

We use a canonical query in graph traversal, namely reachability, for the comparison. Let *distance* d between nodes x and y be the smallest number of edges connecting x and y . For each d , we randomly pick 1000 $\{x, y\}$ pairs and measure the time to determine reachability at distance d between x and y , using Neo4j and GRIPP.

³ <http://dl.acm.org/> Many thanks to Craig Rodkin at ACM HQ for providing the dataset.

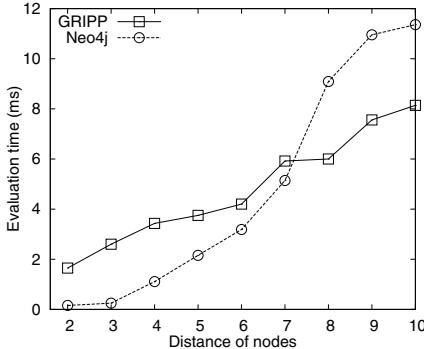


Fig. 2. Comparison of average query time for reachability test

Table 1. ACMDL community statistics over time

time periods	average community size	total number of papers	avg. #coauthors per paper
1975-1980	5	144	2.1
1981-1985	6	356	2.6
1986-1990	6	366	2.9
1991-1995	9	511	4.2
1996-2000	11	535	4.7

Fig. 2 shows that Neo4j is indeed faster for small distances. However, its query time accelerates as d increases because Neo4j uses the Dijkstra algorithm (which has quadratic time complexity). In contrast, traversal time with GRIPP scales linearly. It is even faster than Neo4j for large distances, which is an advantage for large datasets.

5.1 Community Structure

We now use ACMDL to demonstrate how, with sonSchema, community definition is easy and can provide insight into community dynamics.

Community Definition. We illustrate the ease of community definition (Sec. 4.1) with this question: *How do coauthor communities evolve over time?*

Some author lists include people who are actually not active in the research collaboration. To help exclude such authors, we define a coauthor community C to include only those who coauthored at least two papers with others in C .

We select only papers in SIGMOD conferences and divide time into 5-year periods. Table 1 shows that the average community size increases with time. To understand this increase, we run queries to count the number of papers and coauthors. Table 1 shows that the average number of coauthors per paper also increases over time. The insight we get is: *coauthor communities get larger because there are more authors per paper.*

A sample query to get the number of coauthors in a time period is:

```
select count(*) from coauthor, proceedings p, conference c
where coauthor.paper_id=p.paper_id
and p.proceeding_id=c.proceeding_id
and year(c.publication_date)>1995
and year(c.publication_date)<=2000
and c.proc_profile like '%SIGMOD%'
```

Note the use of aggregation (count), joins, selection (year) and non-key attributes (SIGMOD). Such a multi-dimensional analysis using a combination of operators is easy for SQL, but hard to formulate as a query on any graph-based model of coauthorship.

Community Growth. Sec. 4.1 describes how one can study the growth of a community C by measuring the probability that someone in $\text{fringe}(C)$ later joins C . We now demonstrate how sonSchema facilitates the analysis of social dynamics with this question: *How does coauthorship history affect the joining probability?*

A graph metric often used for cluster discovery is conductance [10], whose value is smaller for a tighter community (see Sec. 5.2). Analysis via sonSchema complements such graph-based analysis; we illustrate this now by choosing 10 communities from year 1999 that have conductance smaller than 0.06, and size from 4 to 8 authors. For each community C , we define $\text{fringe}(C)$ as those authors x not in C but have written a paper with someone in C in the last 10 years (i.e. 1990–1999). These x are further classified into: **Class (I)** 0 coauthorship in 1997–1999; **Class (II)** 1 to 3 coauthorships in 1997–1999; **Class (III)** 4 or more coauthorships in 1997–1999.

For each class, we then compute $\text{prob}(C|K, 1 \text{ year})$ that an x in $\text{fringe}(C)$, who has more than K coauthorships with members of C over the last 10 years, joins C between 1999 and 2000. For $K = 8$, the experiments find that:

- A Class(I) author x , i.e. who has 0 coauthorship with C in 1997–1999, has a low probability 0.16 of joining C , even though x has more than 8 coauthorships with C .
- A Class(III) author x , i.e. who has 4 or more coauthorships with C in 1997–1999, has a high probability 0.94 of joining C .

(For Class(II), the probability is 0.74.) The insight we get is: *recent coauthorship affects $\text{prob}(C|K, 1 \text{ year})$ more than a high coauthorship count (that is spread over 10 years).*

Again, extracting such an insight involving time and aggregation, etc., is easy with sonSchema, but difficult if ACMDL is modeled as a graph.

5.2 Cluster Discovery

To evaluate our SQL-based Algo. 1 for cluster discovery, we compare it to the heuristic-based hierarchical decomposition [6] and the approximation-based local spectral algorithm [12], which represent the two major graph-based discovery techniques [13]. Our Algo. 1 uses the author with the highest coauthorship frequency to seed each cluster, so we rank the clusters in the order that they are generated. An empirical comparison of current techniques found that minimizing conductance produces better clusters [13], so we use conductance to rank (smaller is better) clusters for the graph algorithms.

Table 2. Number of authors in a cluster (excluding trivial clusters of size 1 and 2)

	maximum		minimum		average	
algo.	all	top-20	all	top-20	all	top-20
SQL	34	24	3	3	5	9
heuristic	2358	112	3	32	6	51
approx.	7733	150	3	46	1452	105

Table 3. Number of coauthorships in a cluster

	maximum		minimum		average	
algo.	all	top-20	all	top-20	all	top-20
SQL	296	208	4	4	17	75
heuristic	8826	247	2	69	16	164
approx.	20251	505	3	203	3925	355

Table 4. Number of coauthorships per author in a cluster

	maximum		minimum		average	
algo.	all	top-20	all	top-20	all	top-20
SQL	20.3	16.9	1.3	1.3	3.1	8.9
heuristic	14.0	5.2	0.7	1.7	1.6	3.3
approx.	6.7	5.2	1.0	3.0	2.7	3.5

Table 5. Conductance of a cluster

	maximum		minimum		average	
algo.	all	top-20	all	top-20	all	top-20
SQL	0.96	0.96	0.008	0.024	0.381	0.556
heuristic	0.20	0.00	0.000	0.000	0.001	0.000
approx.	0.14	0.01	0.002	0.003	0.014	0.006

To compare the algorithms, we use data from the data mining community (identified by the keywords, title and abstract of each paper in ACM DL). The choice of threshold τ and stop condition for Algo. 1 are as mentioned in Sec. 4.2.

Table 2 shows that the heuristic- and approximation-based techniques generate big clusters: their minimum top-20 cluster sizes are about 30-50, and the approximation-based technique has average cluster size of 1452 authors.

Besides size, we also examine the *quality* of the clusters. Table 3 shows that the number of coauthorships per cluster is large for the approximation-based technique. Moreover, it generates many clusters of more than 1000 coauthorships each, and the average cluster has 3925 coauthorships; these numbers are arguably too high for the clustering to be meaningful. Table 4 further shows that our SQL-based technique is significantly better in terms of coauthorships per author. For example, the heuristic-based clusters have an average of only 1.6 coauthorships per author, partly because they have bigger sizes (Table 2).

Algo. 1 thus identifies clusters that are smaller in size and higher in quality. Yet, as Table 5 shows, the clusters found by the graph algorithms have very small conductance. It also shows that the SQL-based clusters have very high conductance. The insight we get is: *Conductance minimization is neither sufficient nor necessary for discovering good clusters.*

Finally, we compare the size distribution for Algo. 1 and the heuristic-based technique. (We omit the approximation-based technique because it gives just one cluster for each size [13].) Fig. 3 shows that the heuristic-based technique generates many small clusters (5083 clusters of 2 authors each) and another 7341 clusters that include huge ones (up to 2358 authors). In contrast, Algo. 1 finds 2814 clusters (none with size 1 or 2), the largest of which has 34 authors. Thus, Algo. 1 divides the social network into a small number of congenial clusters, whereas the heuristic-based technique produces many clusters that have unfriendly size (too small or too big).

Note that one can easily modify Algo. 1 to use some other preferred metrics.

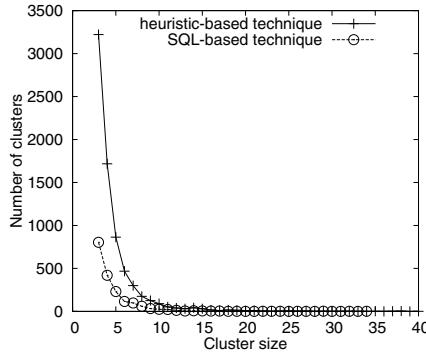


Fig. 3. Comparing cluster size distribution. (The heuristic-based technique generates 5083 clusters of size 2, which are not shown here.)

Table 6. Statistics for Twitter dataset

time period	2008-11-11 to 2009-11-06
number of users	456107
maximum/average/minimum number of followers	500/332/1
maximum/average/minimum number of people that a user follows	198/4/1
number of followed-follower pairs	815211
number of user pairs who follow each other	2494
number of tweets	28688584
number of tweets that are original tweets	26161245
number of tweets that are retweets	498991
number of users who have posted or replied to at least one tweet	274315
maximum/average/minimum number of tweets that are a user's original post	383/95.5/1
maximum/average/minimum number of tweets that are a user's reply	200/12.6/1
maximum/average/minimum number of tweets that are just retweets	195/4.4/1

5.3 Action Propagation

We now demonstrate how our sonSchema model of Twitter data (Sec. 3.1) can support an analysis of action propagation (Sec. 4.3); statistics for the Twitter dataset are given in Table 6. Consider this question: *Does the number of followers a user has affect how far her tweet propagates?*

We first prepare a table *retweet* that stores all retweet information for each tweet up to 4 hops. We then group users according to how many followers they have, as shown in Table 7; there are 352 users who each has 1–50 followers, 161 users who each has 51–100 followers, etc. To compute Table 7, we first compute table *follower_cnt* that stores *user_id* and her number of followers, and table *follower_nhop* that stores *user_id* and her followers up to 3 hops. We then run the query

```
select fc.uid, count(rt.tid)
from follower_cnt fc, follower_nhop fn, retweet rt
where fn.uid = fc.uid and fc.uid=rt.uid
and fc.f_count≥1 and fc.f_count≤50
and rt.ruid=fn.fid and fn.hop=1
group by fc.uid
```

Table 7. Breadth of following: 352 users who each has 1–50 followers, etc

n	1–50	51–100	101–200	201–300	301–400	401–500
#users with n followers	352	161	242	147	141	1433

Table 8. Depth of propagation: for a user u with 201–300 followers, 0.5% of retweets of u 's messages were by 3-hop followers, etc

K -hop	range in number of followers					
	1–50	51–100	101–200	201–300	301–400	401–500
1	97.7%	97.9%	98%	97.6%	98.5%	99.4%
2	2.3%	1.8%	1.7%	1.9%	1.4%	0.4%
3	0%	0.3%	0.3%	0.5%	0.1%	0.1%
≥ 4	0%	0%	0%	0%	0%	0%

Computation for other ranges is similar. This query illustrates the need for several joins and aggregation in multiple dimensions (followers and retweets) if one is to gain insight into network dynamics.

Table 8 presents our query results on propagation depth. Consider the row for 3 hops: It shows that the 3-hop penetration increases from 0% for a user with 1–50 followers to 0.5% for one with 201–300 followers. Beyond that, the 3-hop penetration actually drops for users with more followers. This shows that it is not true that the more followers u has, the farther u 's messages will propagate, i.e. the insight we get is: *breadth of following does not determine depth of penetration*.

This observation would be relevant to merchants who are considering coupon dissemination via social networks (Sec. 4.3).

6 Related Work

Most of the related work were already cited above. We now mention some others.

To see the novelty in sonSchema, one can compare it to the schemas for SoQL [17] and NetIntel [24]; they consist of just nodes, edges, node types and edge types. In contrast, sonSchema explicitly models social products and interactions, as well as user-product and product-product relationships.

A survey of the literature on community structure [3,12,20], link prediction [11,15] and social influence [9,22] shows that graphs are the predominant model for social networks. However, a social graph suffices for some of these studies because they do not analyze the interactions (which we do through **response2post** and **social_product_activity**).

Cluster discovery techniques can be classified under heuristics and approximations [13]; Sec. 4.2 therefore compares our SQL-based technique to a representative of each.

Several papers have studied DBLP as a social network [3,26]. Instead, we use ACM DL, which is much richer in terms of information (citation, affiliation, keywords, abstracts, etc.). Goyal et al.'s study of action propagation [9] requires an action log for Flickr, which we do not have. Instead, we use data from Twitter (with *tweet* as *Action*).

7 Current Work

We can use sonSchema's restricted form to re-engineer MySQL for scalability. E.g. we believe it is possible to incorporate its schema graph into a concurrency control and thus provide strong consistency, but without the ACID bottleneck [21].

For now, we are studying the structure that sonSchema imposes on the space of all join trees. This study may identify bushy strategies for multi-way joins that execute faster than strategies that are produced by current optimizers [7].

References

1. Andersen, R., Chung, F., Lang, K.: Local graph partitioning using pagerank vectors. In: FOCS, pp. 475–486 (2006)
2. Angles, R., Gutiérrez, C.: Survey of graph database models. Comput. Surv. 40(1) (2008)
3. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: membership, growth, and evolution. In: Proc. KDD, pp. 44–54 (2006)
4. Bao, Z., Tay, Y.C., Zhou, J.: A conceptual schema for social networks, <http://sonsql.comp.nus.edu.sg/rsn.pdf>
5. Bao, Z., Zeng, Y., Tay, Y.C.: sonLP: Social network link prediction by principal component regression. In: Proc. ASONAM (to appear, 2013)
6. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E, 1–6 (2004)
7. Cluet, S., Moerkotte, G.: On the complexity of generating optimal left-deep processing trees with cross products. In: Vardi, M.Y., Gottlob, G. (eds.) ICDT 1995. LNCS, vol. 893, pp. 54–67. Springer, Heidelberg (1995)
8. Flake, G.W., Tarjan, R.E., Tsoutsouliklis, K.: Graph clustering and minimum cut trees. Internet Mathematics 1(4) (2003)
9. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Learning influence probabilities in social networks. In: WSDM, pp. 241–250 (2010)
10. Kannan, R., Vempala, S., Vetta, A.: On clusterings: Good, bad and spectral. J. ACM 51(3), 497–515 (2004)
11. Kashima, H., Abe, N.: A parameterized probabilistic model of network evolution for supervised link prediction. In: Proc. ICDM, pp. 340–349 (2006)
12. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: WWW, pp. 695–704 (2008)
13. Leskovec, J., Lang, K.J., Mahoney, M.: Empirical comparison of algorithms for network community detection. In: Proc. WWW, pp. 631–640 (2010)
14. Levene, M., Loizou, G.: Why is the snowflake schema a good data warehouse design? Inf. Syst. 28(3), 225–240 (2003)
15. Liben-Nowell, D., Kleinberg, J.M.: The link-prediction problem for social networks. JASIST 58(7), 1019–1031 (2007)
16. Lichtenwalter, R.N., Lussier, J.T., Chawla, N.V.: New perspectives and methods in link prediction. In: Proc. KDD, pp. 243–252 (2010)
17. Ronen, R., Shmueli, O.: SoQL: A language for querying and creating data in social networks. In: Proc. ICDE, pp. 1595–1602 (2009)
18. Rys, M.: Scalable SQL. Commun. ACM 54(6), 48–53 (2011)
19. Shakkottai, S., Ying, L., Sah, S.: Targeted coupon distribution using social networks. SIGMETRICS Perf. Eval. Rev. 38, 26–30 (2011)

20. Spiliopoulou, M.: Evolution in social networks: A survey. In: Social Network Data Analytics, pp. 149–175. Springer (2011)
21. Stonebraker, M., Cattell, R.: 10 rules for scalable performance in ‘simple operation’ databases. Commun. ACM 54, 72–80 (2011)
22. Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: Proc. KDD, pp. 807–816 (2009)
23. Trißl, S., Leser, U.: Fast and practical indexing and querying of very large graphs. In: Proc. SIGMOD, pp. 845–856 (2007)
24. Tsvetovat, M., Diesner, J., Carley, K.: NetIntel: A database for manipulation of rich social network data. Technical Report CMU-ISRI-04-135, Carnegie Mellon University (2005)
25. Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P., Zhao, B.Y.: User interactions in social networks and their implications. In: Proc. EuroSys, pp. 205–218 (2009)
26. Zaiane, O.R., Chen, J., Goebel, R.: DBconnect: mining research community on DBLP data. In: Proc. WebKDD/SNA-KDD, pp. 74–81 (2007)

Minimizing Human Effort in Reconciling Match Networks

Hung Quoc Viet Nguyen¹, Tri Kurniawan Wijaya¹, Zoltán Miklós², Karl Aberer¹, Eliezer Levy³, Victor Shafran³, Avigdor Gal⁴, and Matthias Weidlich⁴

¹ École Polytechnique Fédérale de Lausanne

² Université de Rennes 1

³ SAP Research Israel

⁴ Technion – Israel Institute of Technology

Abstract. Schema and ontology matching is a process of establishing correspondences between schema attributes and ontology concepts, for the purpose of data integration. Various commercial and academic tools have been developed to support this task. These tools provide impressive results on some datasets. However, as the matching is inherently uncertain, the developed heuristic techniques give rise to results that are not completely correct. In practice, post-matching human expert effort is needed to obtain a correct set of correspondences. We study this post-matching phase with the goal of reducing the costly human effort. We formally model this human-assisted phase and introduce a process of *matching reconciliation* that incrementally leads to identifying the correct correspondences. We achieve the goal of reducing the involved human effort by exploiting a *network* of schemas that are matched against each other. We express the fundamental matching constraints present in the network in a declarative formalism, Answer Set Programming that in turn enables to reason about necessary user input. We demonstrate empirically that our reasoning and heuristic techniques can indeed substantially reduce the necessary human involvement.

1 Introduction

Schema and ontology matching is the process of establishing correspondences between the attributes and ontology concepts, for the purpose of data integration. There is a large body of work on techniques to support the schema and ontology matching task and numerous commercial and academic tools, called matchers, have been developed in recent years [4,21,10]. Even though those matchers achieve impressive performance on some datasets, they cannot be expected to yield a completely correct result since they rely on heuristic techniques. In practice, data integration tasks often include a post-matching phase, in which correspondences are reviewed and validated by experts.

User involvement in an integration process is still an open challenge [23], with the main difficulty being the design of a burdenless post-matching phase. In this work we propose a method, in which a matcher introduces a user with carefully selected correspondences, aiming at the minimization of user involvement in the integration process. The proposed algorithm is based on automatic constraint violation detection using Answer Set Programming (ASP).

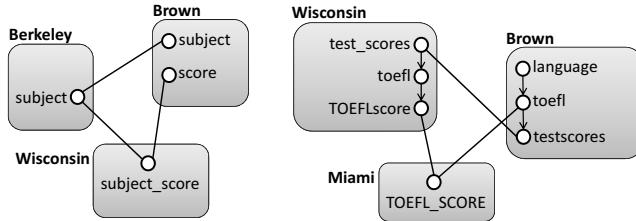


Fig. 1. Violations of network-level constraints in real-world Web schemas

We focus on a setting in which matching is conducted for a network of related schemas that are matched against each other. In this setting, we streamline the potentially chaotic post-matching phase as a structured reconciliation of conflicting correspondences in the network. The notion of matching networks is interesting in its own right and is beneficial in many real world scenarios, see for example [24]. In principle, a matching network enables collaborative integration scenarios, and scenarios where a monolithic mediated schema approach is too costly or simply infeasible. In our work, having a network of multiple schemas enables introducing network-level consistency constraints, thereby increasing the potential for guided improvement of the matching process. Namely, we go beyond the common practice of improving and validating matchings by considering a pair of schemas. Instead, we consider network-level consistency constraints as a means to improve the matchings over the entire network. We consider network-level constraints that are fundamental, domain-independent and, therefore, are expected to hold universally in any matching network. The process of network reconciliation automatically detects constraint violations in the network, and minimizes the user effort by identifying the most critical violations that require user feedback, thus supporting and guiding a human user in the post-matching phase.

In constructing the proposed mechanism we separate the task into two parts. The first, which can be considered a “design time” process, involves the encoding of constraints the violation of which indicates a “suspicious” correspondence. Such constraints are defined in a meta-level to be applied to newly introduced schemas. This process requires highly skilled matching specialists and we show several examples of such constraints in this work. The “run-time” part of the process involves an interactive system through which a user is introduced with correspondences that violate the pre-defined constraints. Such correspondences are validated by the domain expert, followed by a new iteration of constraint violation detection.

As an illustrating example, consider the matching of university application forms. A schema matching network is created by establishing pairwise matchings between the schemas. Clearly, these matchings shall obey certain constraints. The pairwise schema matching, for instance, often ensures the 1 : 1 correspondence constraint: each attribute corresponds to at most a single attribute in another schema.

Figure 1 illustrates the notion of network-level constraints we investigate in this work. Figure 1(left) introduces a violation of a *cycle constraint*: If the matchings form a cycle across multiple schemas, then the attribute correspondences should form a closed cycle. In our example, *subject* in the *Brown* schema is connected to another attribute *score*

of the same schema by a network-wide path of correspondences. Figure 1(right) illustrates another constraint violation based on a directed relation between attributes within a schema, for instance, a composition relation in an XML schema. The *dependency constraint* requires that the relation between attributes within a schema is preserved by network-wide paths of correspondences. This constraint is satisfied for the pairwise matching between schemas *Wisconsin* and *Brown*. In contrast, it is violated on the network level, once schema *Miami* is taken into account.

Detecting and eliminating violations of such constraints is a tedious task because of the sheer amount of violations usually observed and the need to inspect large parts of the matching network. Our techniques can automatically detect such constraint violations, minimize the necessary feedback steps for resolving the violations, as well as detect erroneous feedback in most cases. The main contribution of this work can be summarized as follows:

- We introduce the concept of matching networks, a generalization of the pairwise schema matching setting.
- We define a model for reconciliation in matching networks.
- We present a framework that allows expressing generic matching constraints in a declarative form, using Answer Set Programs (ASP). This expressive declarative formalism enables us to formulate a rich set of matching constraints.
- We propose a heuristic for validation ordering in a schema network setting.
- We conducted experiments with real-world schemas, showing that the proposed algorithm outperforms naïve methods of choosing correspondences for feedback gathering, reducing user input by up to 40%.

The rest of the paper is organized as follows. The next section introduces our model and formalizes the addressed problem. Section 3 shows how reconciliation of matching networks is implemented using answer set programming. Using this framework, we propose approaches to minimize human involvement in Section 4. Report on our empirical evaluation is given in Section 5. Finally, we review our contribution in the light of related work in Section 6, before Section 7 concludes the paper.

2 Model and Problem Statement

In this section, we introduce matching networks (Section 2.1), a model of the reconciliation process (Section 2.2), and the reconciliation problem definition (Section 2.3).

2.1 Matching Networks

A schema $s = (A_s, \delta_s)$ is a pair, where $A_s = \{a_1, \dots, a_n\}$ is a finite set of *attributes* and $\delta_s \subseteq A_s \times A_s$ is a relation capturing *attribute dependencies*. This model largely abstracts from the peculiarities of schema definition formalisms, such as relational or XML-based models. As such, we do not impose specific assumptions on δ_s , which may capture different kinds of dependencies, *e.g.*, composition or specialization of attributes.

Let $\mathcal{S} = \{s_1, \dots, s_n\}$ be a set of schemas that are built of unique attributes ($\forall 1 \leq i \neq j \leq n, A_{s_i} \cap A_{s_j} = \emptyset$) and let $A_{\mathcal{S}}$ denote the set of attributes in \mathcal{S} , *i.e.*, $A_{\mathcal{S}} = \bigcup_i A_{s_i}$. The *interaction graph* $G_{\mathcal{S}}$ represents which schemas need to be matched in the network.

Therefore, the vertices in $V(G_S)$ are labeled by the schemas from \mathcal{S} and there is an edge between two vertices, if the corresponding schemas need to be matched.

An *attribute correspondence* between a pair of schemas $s_1, s_2 \in \mathcal{S}$ is an attribute pair $\{a, b\}$, such that $a \in A_{s_1}$ and $b \in A_{s_2}$. A *valuation function* associates a value in $[0, 1]$ to an attribute correspondence. *Candidate correspondences* $c_{i,j}$ (for a given pair of schemas $s_i, s_j \in \mathcal{S}$) is a set of attribute correspondences, often consisting of correspondences whose associated value is above a given threshold. The set of candidate correspondences C for an interaction graph G_S consists of all candidates for pairs corresponding to its edges, i.e. $C = \bigcup_{(s_i, s_j) \in E(G_S)} c_{i,j}$. C is typically the outcome of first-line schema matchers [12]. Most such matchers generate simple 1 : 1 attribute correspondences, which relate an attribute of one schema to at most one attribute in another schema. In what follows, we restrict ourselves to 1 : 1 candidate correspondences for simplicity sake. Extending the proposed framework to more complex correspondences can use tools that were proposed in the literature, e.g., [13].

A *schema matching* for G_S is a set D of attribute correspondences $D \subseteq C$. Such schema matching is typically generated by second-line matchers, combined with human validation, and should adhere to a set of predefined constraints $\Gamma = \{\gamma_1, \dots, \gamma_n\}$. Such constraints may require, for example, that at least 80% of all attributes are matched. A schema matching D is *valid* if it satisfies all of the constraints in Γ .

Combining the introduced notions, we define a *matching network* to be a quadruple $(\mathcal{S}, G_S, \Gamma, C)$, where \mathcal{S} is a set of schemas (of unique attributes), G_S a corresponding interaction graph, Γ a set of constraints, and C a set of candidate correspondences.

2.2 Reconciliation Process

The set of candidate correspondences C aims at serving as a starting point of the matching process and typically violates the matching constraint set Γ . In this section, we model the reconciliation process under a set of predefined constraints Γ (see Section 3) as an iterative process, where in each step a user asserts the correctness of a single correspondence. Starting with the result of a matcher, a set of correspondences, called an *active set*, is continuously updated by: (1) selecting an attribute correspondence $c \in C$, (2) eliciting user input (approval or disapproval) on the correspondence c , and (3) computing the consequences of the feedback and updating the active set. Reconciliation halts once the goal of reconciliation (e.g., eliminating all constraint violations) is reached. It is worth noting that in general, a user may add missing correspondences to C during the process. For simplicity, we assume here that all relevant candidate correspondences are already included in C .

Each user interaction step is characterized by a specific index i . Then, D_i denotes the set of correspondences considered to be true in step i dubbed the *active set*. Further, let $u_c^+ (u_c^-)$ denote the user input where u_c^+ denotes approval and u_c^- denotes disapproval of a given correspondence $c \in C$ and $U_C = \{u_c^+, u_c^- \mid c \in C\}$ be the set of all possible user inputs for the set of correspondences C . Further, $u_i \in U_C$ denotes user input at step i and $U_i = \{u_j \mid 0 \leq j \leq i\}$ is the set of user input assertions until step i . The consequences of such user input assertions U_i are modeled as a set $Cons(U_i) \subseteq U_C$ of positive or negative assertions for correspondences. They represent all assertions that can be concluded from the user input assertions.

Algorithm 1. Generic reconciliation procedure

```

input : a set of candidate correspondences  $C$ , a set of constraints  $\Gamma$ , a reconciliation goal  $\Delta$ .
output: the reconciled set of correspondences  $D_r$ .
// Initialization
1  $D_0 \leftarrow C; U_0 \leftarrow \emptyset; Cons(U_0) \leftarrow \emptyset; i \leftarrow 0;$ 
2 while  $not \Delta$  do
    // In each user interaction step (1) Select a correspondence
    3  $c \leftarrow select(C \setminus \{c \mid u_c^+ \in Cons(U_i) \vee u_c^- \in Cons(U_i)\});$ 
    // (2) Elicit user input
    4 Elicit user input  $u_i \in \{u_c^+, u_c^-\}$  on  $c$ ;
    // (3) Integrate the feedback
    5  $U_{i+1} \leftarrow U_i \cup \{u_i\};$ 
    6  $Cons(U_{i+1}) \leftarrow conclude(U_i);$ 
    7  $D_{i+1} \leftarrow D_i \cup \{c \mid u_c^+ \in Cons(U_{i+1})\} \setminus \{c \mid u_c^- \in Cons(U_{i+1})\};$ 
    8  $i \leftarrow i + 1;$ 

```

A generic reconciliation procedure is illustrated in Algorithm 1. It takes a set of candidate correspondences C , a set of constraints Γ , and a reconciliation goal Δ as input and returns a reconciled set of correspondences D_r . Initially (line 1), the active set D_0 is given as the set of candidate correspondences C and the sets of user input U_0 and consequences $Cons(U_0)$ are empty. Then, we proceed as follows: First, there is a function *select*, which selects a correspondence from the set of candidate correspondences (line 3). Here, all correspondences for which we already have information as the consequence of earlier feedback (represented by $Cons(U_i)$) are neglected. Second, we elicit user input for this correspondence (line 4). Then, we integrate the feedback by updating the set of user inputs U_{i+1} (line 5), computing the consequences $Cons(U_{i+1})$ of these inputs with function *conclude* (line 6), and updating the active set D_{i+1} (line 7). A correspondence is added to (removed from) the active set, based on a positive (negative) assertion of the consequence of the feedback. The reconciliation process stops once D_r satisfies the halting condition Δ representing the goal of reconciliation.

Instantiations of Algorithm 1 differ in their implementation of the *select* and *conclude* routines. For example, by considering one correspondence at a time, Algorithm 1 emulates a manual reconciliation process followed by an expert. As a baseline, we consider an expert working without any tool support. This scenario corresponds to instantiating Algorithm 1 with a selection of a random correspondence from $C \setminus Cons(U_i)$ ($select(C \setminus Cons(U_i))$) and the consequences of user input are given by the input assertions U_i ($conclude(U_i) = U_i$).

2.3 Problem Statement

Given the iterative model of reconciliation, we would like to minimize the number of necessary user interaction steps for a given reconciliation goal. Given a schema matching network $(\mathcal{S}, G_S, \Gamma, C)$, a reconciliation goal Δ , and a sequence of correspondence sets $\langle D_0, D_1, \dots, D_n \rangle$ such that $D_0 = C$ (termed a *reconciliation sequence*), we say

that $\langle D_0, D_1, \dots, D_n \rangle$ is *valid* if D_n satisfies Δ . Let \mathcal{R}_Δ denote a finite set of valid reconciliation sequences that can be created by instantiations of Algorithm 1. Then, a reconciliation sequence represented by $\langle D_0, D_1, \dots, D_n \rangle \in \mathcal{R}_\Delta$ is *minimal*, if for any reconciliation sequence $\langle D'_0, D'_1, \dots, D'_m \rangle \in \mathcal{R}_\Delta$ it holds that $n \leq m$.

Our objective is defined in terms of a minimal reconciliation sequence, as follows.

Problem 1. *Let $(\mathcal{S}, G_\mathcal{S}, \Gamma, C)$ be a schema matching network and \mathcal{R}_Δ a set of valid reconciliation sequences for a reconciliation goal Δ . The minimal reconciliation problem is the identification of a minimal sequence $\langle D_0, D_1, \dots, D_n \rangle \in \mathcal{R}_\Delta$.*

Problem 1 is basically about designing a good instantiation of *select* and *conclude* to minimize the number of iterations to reach Δ . The approach taken in this paper strives to reduce the effort needed for reconciliation, thus finding a heuristic solution to the problem. We achieve this goal by relying on heuristics for the selection of correspondences (*select*) and applying reasoning for computing the consequences (*conclude*).

3 Reconciliation Using ASP

This section introduces our tool of choice for instantiating the *select* and *conclude* routines, Answer Set Programming (ASP). ASP is rooted in logic programming and non-monotonic reasoning; in particular, the stable model (answer set) semantics for logic programs [15,16] and default logic [22]. In ASP, solving search problems is reduced to computing answer sets, such that answer set solvers (programs for generating answer sets) are used to perform search. We start by shortly summarizing the essentials of ASP (Section 3.1). We then show how to model schema matching networks, introduced in Section 2.1, using ASP (Section 3.2). Section 3.3 provides three examples of schema matching constraints in ASP. Finally, we outline the reasoning mechanism ASP uses to identify violations of matching constraints (Section 3.4).

3.1 Answer Set Programming

We now give an overview of ASP. Formal semantics for ASP and further details are given in [9]. Let \mathcal{C} , \mathcal{P} , \mathcal{X} be mutually disjoint sets whose elements are called *constant*, *predicate*, and *variable* symbols, respectively. Constant and variable symbols $\mathcal{C} \cup \mathcal{X}$ are jointly referred to as *terms*. An *atom* (or *strongly negated atom*) is defined as a predicate over terms. It is of the form $p(t_1, \dots, t_n)$ (or $\neg p(t_1, \dots, t_n)$, respectively) where $p \in \mathcal{P}$ is a predicate symbol and t_1, \dots, t_n are terms. An atom is called *ground* if t_1, \dots, t_n are constants, and *non-ground* otherwise. Below, we use lower cases for constants and upper cases for variables in order to distinguish both types of terms.

An answer set program consists of a set of disjunctive rules of form:

$$a_1 \vee \dots \vee a_k \leftarrow b_1, \dots, b_m, \dots, \text{not } c_1, \dots, \text{not } c_n$$

where $a_1, \dots, a_k, b_1, \dots, b_m, c_1, \dots, c_n$ ($k, m, n \geq 0$) are *atoms* or *strongly negated atoms*. This rule can be interpreted as an *if-then* statement: if b_1, \dots, b_m are true and c_1, \dots, c_n are false, then we conclude that at least one of a_1, \dots, a_k is true. We call

a_1, \dots, a_k the *head* of the rule, whereas b_1, \dots, b_m and c_1, \dots, c_n are the *body* of the rule. A rule with an empty body is a *fact*, since the head has to be satisfied in any case. A rule with an empty head is a *constraint*; the body should never be satisfied.

Example 1. Π is an answer set program comprising three rules (X being a variable, c being a constant). Program Π defines three predicates p, q, r . The first rule is a *fact* and the third rule denotes a *constraint*. Further, $p(c), r(c)$ are ground atoms, and $p(X), q(X)$, are non-ground atoms:

$$\Pi = \left\{ \begin{array}{l} p(c) \leftarrow \\ q(X) \leftarrow p(X). \\ \quad \leftarrow r(c). \end{array} \right\}$$

Informally, an answer set of a program is a minimal set of ground atoms, i.e., predicates defined only over constants, that satisfies all rules of the program. An example of an answer set of program Π given in Example 1 would be $\{p(c), q(c)\}$.

Finally, we recall the notion of *cautious* and *brave* entailment for ASPs [9]. An ASP Π *cautiously entails* a ground atom a , denoted by $\Pi \models_c a$, if a is satisfied by *all* answer sets of Π . For a set of ground atoms A , $\Pi \models_c A$, if for each $a \in A$ it holds $\Pi \models_c a$. An ASP Π *bravely entails* a ground atom a , denoted by $\Pi \models_b a$, if a is satisfied by *some* answer sets of Π . For a set of ground atoms A , $\Pi \models_b A$, if for each $a \in A$ it holds that some answer set M satisfies a .

3.2 Representing Matching Networks

Let $(\mathcal{S}, G_{\mathcal{S}}, \Gamma, C)$ be a matching network. An ASP $\Pi(i)$, corresponding to the i -th step of the reconciliation process, is constructed from a set of smaller programs that represent the schemas and attributes ($\Pi_{\mathcal{S}}$), the candidate correspondences (Π_C), the active set D_i ($\Pi_D(i)$), the basic assumptions about the setting (Π_{basic}), the constraints (Π_{Γ}), and a special rule that relates the correspondences and constraints Π_{cc} . The program $\Pi(i)$ is the union of the smaller programs $\Pi(i) = \Pi_{\mathcal{S}} \cup \Pi_C \cup \Pi_D(i) \cup \Pi_{basic} \cup \Pi_{\Gamma} \cup \Pi_{cc}$. We focus in the section on the four first programs. The remaining two programs are discussed in Section 3.3.

Schemas and attributes: $\Pi_{\mathcal{S}}$ is a set of ground atoms, one for each attribute and its relation to a schema, and one for each attribute dependency:

$$\Pi_{\mathcal{S}} = \{attr(a, s_i) \mid s_i \in \mathcal{S}, a \in A_{s_i}\} \cup \{dep(a_1, a_2) \mid s_i \in \mathcal{S}, (a_1, a_2) \in \delta_{s_i}\}$$

Candidate correspondences: Π_C comprises ground atoms, one for each candidate correspondence in the matching network: $\Pi_C = \{cor(a_1, a_2) \mid (a_1, a_2) \in C\}$

Active set: $\Pi_D(i)$ is a set of ground atoms, corresponding to the active set D_i :

$$\Pi_D(i) = \{corD(a_1, a_2) \mid (a_1, a_2) \in D_i\}$$

Basic assumptions: rules in Π_{basic} , as follows.

- An attribute cannot occur in more than one schema. We encode this knowledge by adding a rule with an empty head, i.e., a constraint, so that no computed answer set will satisfy the rule body. For each attribute $a \in A_{\mathcal{S}}$ and schemas $s_1, s_2 \in \mathcal{S}$, we add the following rule to Π_{basic} : $\leftarrow attr(a, s_1), attr(a, s_2), s_1 \neq s_2$.

- *There should be no correspondence between attributes of the same schema.* We add a rule to for each candidate correspondence $(a_1, a_2) \in C$ and schemas $s_1, s_2 \in S$ to Π_{basic} : $\leftarrow cor(a_1, a_2), attr(a_1, s_1), attr(a_2, s_2), s_1 = s_2$.
- *The active set is a subset of all matching candidates.* We add a rule to Π_{basic} : $cor(X, Y) \leftarrow corD(X, Y)$.

3.3 Matching Constraints in ASP

Matching constraints are defined to ensure the integrity of the matching process. Such constraints are the subject of research in the schema and ontology matching research area (see Section 6). They are defined independently of the application domain and can be mixed and matched based on the needs of such applications. In this section, we illustrate the modeling of three such constraints using ASP followed by the modeling of the connection between correspondences and constraints.

Constraints (Π_Γ). We express matching constraints as rules in the program Π_Γ , one rule per constraint, such that $\Pi_\Gamma = \Pi_{\gamma_1} \cup \dots \cup \Pi_{\gamma_n}$ for $\Gamma = \{\gamma_1, \dots, \gamma_n\}$. In the following, we give examples of three matching constraints.

- *1 : 1 constraint:* Any attribute of one schema has at most one corresponding attribute in another schema. We capture this constraint with the following rule:
 $\leftarrow match(X, Y), match(X, Z), attr(Y, S), attr(Z, S), Y \neq Z$.
- *Cycle constraint:* Two different attributes of a schema must not be connected by a path of matches. We call a cycle of attribute correspondences *incorrect*, if it connects two different attributes of the same schema, see Figure 1(left) for example. Formally, a solution is valid if it does not contain any incorrect cycles. We encode this constraint based on a reachability relation (represented by $reach(X, Y)$, where X and Y are variables representing attributes) as follows:

$$\begin{aligned} reach(X, Y) &\leftarrow match(X, Y) \\ reach(X, Z) &\leftarrow reach(X, Y), match(Y, Z) \\ &\quad \leftarrow reach(X, Y), attr(X, S), attr(Y, S), X \neq Y. \end{aligned}$$

- *Dependency constraint:* Dependencies between attributes shall be preserved by paths of matches. To encode this type of constraint, we proceed as follows. First, we model (direct or indirect) reachability of two attributes in terms of the dependency relation (represented by $reachDep(X, Y)$, where X and Y are both variables representing attributes). Then, we require that reachability based on the *match* relation for two pairs of attributes preserves the reachability in terms of the dependency relation between the attributes of either schema:

$$\begin{aligned} reachDep(X, Y) &\leftarrow dep(X, Y) \\ reachDep(X, Z) &\leftarrow reachDep(X, Y), dep(Y, Z) \\ &\quad \leftarrow reachDep(X, Y), reach(X, B), \\ &\quad \quad \quad reachDep(A, B), reach(Y, A). \end{aligned}$$

The choice of constraints depend on the application at hand. For example, 1 : 1 constraints, while being a common constraint in schema matching applications, may not be relevant to some applications. Also, some constraints may not be a major issue

in certain domains. For example, as part of our empirical evaluation, we have tested the number of cycle constraint violations for five datasets that are different in their characterizations (see Table 1 in Section 5). While there were violations of the cycle constraint in all datasets, ranging from hundreds to tens of thousands violations) there was no clear correlation between the network size (in terms of number of schemas, number of attributes, *etc.*) and the number of violations.

Connecting correspondences and constraints (Π_{cc}). A rule that computes a set of correspondences that satisfy the constraints of the matching network uses a *rule with a disjunctive head*. We encode a *match* relation (represented by $match(X, Y)$) to compute this set. A candidate correspondence $cor(X, Y)$ is either present in or absent from *match*, the latter is denoted as $noMatch(X, Y)$. This is captured by the rule:

$$match(X, Y) \vee noMatch(X, Y) \leftarrow corD(X, Y).$$

3.4 Detecting Constraint Violations

Adopting the introduced representation enables us to compute violations of constraints automatically, with the help of ASP solvers. In large matching networks, detecting such constraint violations is far from trivial and an automatic support is crucial.

We say that a set of correspondences $C' = \{c_1, \dots, c_k\} \subseteq C$ violates a constraint $\gamma \in \Gamma$ if $\Pi_S \cup \Pi_{basic} \cup \Pi_\gamma \not\models_b \Pi_{C'}$. In practice, we are not interested in all possible violations, but rather the minimal ones, where a set of violations is minimal w.r.t. γ if none of its subsets violates γ . Given a set of correspondences C' , we denote the set of minimal violations as

$$Violation(C') = \{C'' \mid C'' \subseteq C', \Pi_S \cup \Pi_{basic} \cup \Pi_\gamma \not\models_b C'', \gamma \in \Gamma, C'' \text{ is minimal}\}.$$

The ASP representation also allows for expressing reconciliation goals. A frequent goal of experts is to eliminate all violations: $\Delta_{NoViol} = \{\Pi(i) \models_b \Pi_D(i)\}$, i.e., the joint ASP bravely entails the program of the active set.

4 Minimizing User Effort

The generic reconciliation process (Section 2.2) comprises three steps, namely correspondence selection, user input elicitation, and feedback integration. We argued that this process is characterized by two functions in particular, namely *select* and *conclude*. We now suggest a specific implementation of the two functions. In Section 4.1, we show how to use reasoning techniques to derive consequences for user input. In Section 4.2, we discuss heuristic ordering strategies for correspondence selection.

4.1 Effort Minimization by Reasoning

In the baseline reconciliation process, the consequences of the user input U_i up to step i of the reconciliation process are directly given by the respective input assertions, i.e., $Cons(U_i) = U_i$. This means that updating the active set of correspondences D_i based on $Cons(U_i)$ considers only correspondences for which user input has been elicited.

In the presence of matching constraints, however, we can provide more efficient ways to update the active set. Due to space considerations, we do not detail this process. Instead, we illustrate it using the following example.

Example 2 (Reasoning with user input). Consider two schemas, s_1 and s_2 , and three of their attributes, encoded in ASP as $attr(x, s_1)$, $attr(y, s_2)$, and $attr(z, s_2)$. Assume that a matcher generated candidate correspondences that are encoded as $C = \{cor(x, y), cor(x, z)\}$. Further, assume that Γ consists of the $1 : 1$ constraint. By approving correspondence (x, y) , we can conclude that candidate correspondence (x, z) must be false and should not be included in any of the answer sets. Hence, in addition to validation of correspondence (x, y) , falsification of correspondence (x, z) is also a consequence of the user input on (x, y) .

4.2 Effort Minimization by Ordering

We now consider minimization of user effort based on the selection strategy that is used for identifying the correspondence that should be presented to the user. In Section 2.2, we showed that without any tool support, a random correspondence would be chosen. Depending on the order of steps in the reconciliation process, however, the number of necessary input steps might vary. Some input sequences may help to reduce the required user feedback more efficiently. In this section, we focus on a heuristic selection strategy that exploits a ranking of correspondences for which feedback shall be elicited.

Our selection function is based on a *min-violation scoring* that refers to the number of violations that are caused by a correspondence. The intuition behind this heuristic is that eliciting feedback on correspondences that violate a high number of constraints is particular beneficial for reconciliation of a matching network. As defined in Algorithm 1, selection is applied to the set of candidate correspondences C once all correspondences for which we already have information as the consequence of earlier feedback (represented by $Cons(U_i)$) have been removed. In case there are multiple correspondences that qualify to be selected, we randomly choose one.

5 Empirical Evaluation

This section introduces preliminary empirical evaluation. For our evaluation, we used five real-world datasets spanning various application domains, from classical Web form integration to enterprise schemas. All datasets are publicly available¹ and descriptive statistics for the schemas are given in Table 1.

Business Partner (BP): Three enterprise schemas, originally from SAP, which model business partners in SAP ERP, SAP MDM, and SAP CRM systems.

PurchaseOrder (PO): Purchase order e-business documents from various resources.

University Application Form (UAF): Schemas from Web interfaces of American university application forms.

¹ BP, PO, UAF, WebForm are available at http://lsirwww.epfl.ch/schema_matching and Thalia can be found at: <http://www.cise.ufl.edu/research/dbintegrate/thalia/>

WebForm: Automatically extraction of schemas from Web forms of seven different domains (e.g., betting and book shops) using OntoBuilder.²

Thalia: Schemas describing university courses. This dataset has no exact match, and is mainly used in the first experiment concerning constraint violations.

We used two schema matchers, COMA [7] and Auto Mapping Core (AMC) [19]. Reasoning was conducted with the DLV system,³ release 2010-10-14, a state-of-the-art ASP interpreter. All experiments ran on an Intel Core i7 system (2.8GHz, 4GB RAM).

We evaluated our reconciliation framework in different settings. We varied the construction of schema matching networks in terms of dataset, matcher, and network topology. For the reconciliation process, we considered different types of users and reconciliation goals. We measured the quality improvements achieved by reconciliation and the required human efforts as follows:

Precision We measures quality improvement where precision of the active set at step i is defined as $P_i = (|D_i \cap G|)/|D_i|$, with G being the exact match.

User effort is measured in terms of feedback steps relative to the size of the matcher output C , i.e., $E_i = i/|C|$ (where a user examines one correspondence at a time).

Table 1. Statistics for datasets

Dataset	#Schemas	#Attributes/Schema Min./Max.
BP	3	80/106
PO	10	35/408
UAF	15	65/228
WebForm	89	10/120
Thalia	44	3/18

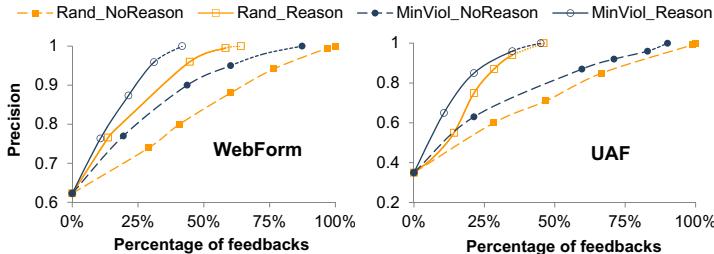


Fig. 2. User effort needed to achieve 100% precision

We studied the extent to which our approach reduces human effort in terms of necessary user feedback steps as follows. For each dataset, we obtained candidate correspondences using COMA. We generated a complete interaction graph and required the 1 : 1 and cycle constraints to hold. Then, we simulated user feedback using the exact matches for the dataset. The reconciliation process starts with the matching results, as determined by COMA.

We explored how the quality of the match result in terms of precision improved when eliciting user feedback according to different strategies. For the WebForm and UAF datasets, Figure 2 depicts the improvements in precision (Y-axis) with increased

² <http://ontobuilder.bitbucket.org/>

³ <http://www.dlvsystem.com>

feedback percentage (X-axis, out of the total number of correspondences) using four strategies, namely

- (1) *Rand_NoReason*: feedback in random order, consequences of feedback are defined as the user input assertions (the baseline described in Section 2.2);
- (2) *Rand_Reason*: reconciliation using random selection of correspondences, but applying reasoning to conclude consequences;
- (3) *MinViol_NoReason*: reconciliation selection of correspondences based on ordering, consequences of feedback are defined as the user input assertions; and finally
- (4) *MinViol_Reason*: reconciliation with the combination of ordering and reasoning for concluding consequences.

The results depicted in Figure 2 show the average over 50 experiment runs. The dotted line in the last segment of each line represents the situation where no correspondence in the active set violated any constraints, *i.e.*, the reconciliation goal Δ_{NoViol} has been reached. In those cases, we used random selection for the remaining correspondences until we reached a precision of 100%. The other datasets (BP and PO) demonstrate similar results and are omitted for brevity sake.

The results show a significant reduction of user effort for all strategies with respect to the baseline. Our results further reveal that most improvements are achieved by applying reasoning to conclude on the consequences of user input. Applying ordering for selecting correspondences provides additional benefits. The combined strategy (*MinViol_Reason*) showed the highest potential to reduce human effort, requiring only 40% or less of the user interaction steps of the baseline.

So far, we assumed that it is always possible to elicit a user input assertion for a correspondence. One may argue that in many practical scenarios, however, this assumption does not hold. Users have often only partial knowledge of a domain, which means that for some correspondences a user cannot provide any feedback. We studied the performance of our approach in this setting, by including the possibility of skipping a correspondence in the reconciliation process. Thus, for certain correspondences, we never elicit any feedback. However, the application of reasoning may allow us to conclude on the assertions for these correspondences as consequences of the remaining user input.

In our experiments, we used a probability p for skipping a correspondence and measured the ratio of concluded assertions (related to skipped correspondences that can be concluded by reasoning over the remaining user input) and all skipped correspondences. Table 2 shows the obtained results. It is worth noting that even with $p = 30\%$, the ratio is close to 0.2, which means that about 20% of the assertions that could not be elicited from the user were recovered by reasoning in the reconciliation process. As expected, this ratio increases as p decreases; skipping less correspondences provides the reasoning mechanism with more useful information.

Table 2. Ability to conclude assertions

Dataset	p : skipping probability					
	5%	10%	15%	20%	25%	30%
BP	0.29	0.26	0.27	0.23	0.20	0.18
PO	0.31	0.30	0.26	0.22	0.22	0.16
UAF	0.21	0.20	0.16	0.15	0.14	0.11
WebForm	0.31	0.32	0.26	0.19	0.16	0.20

6 Related Work

The area of schema and ontology matching was introduced in Section 1. Here, we focus on further work related to two aspects, namely user feedback and constraint validation.

The post-matching phase typically involves human expertise feedback. Several methods for measuring post-matching user effort were proposed in the literature including *overall* [6] and the work of Duchateau et al. [8]. In our work we use a simple measure of user feedback iterations and develop a reconciliation framework to minimize it.

User feedback was proposed for validating query results. Jeffery et al. [17] suggest to establish initial, potentially erroneous correspondences, to be improved through user input. They use a decision-theoretic approach, based on probabilistic matching to optimize the expected effects. FICSR [20] obtains user feedback in the form of query result ranking, taking into account matching constraints. Other works that also focus on data inconsistency include [26] and [14]. Our work focuses on improving correspondences at the metadata level, rather than data.

User feedback on matching through crowd sourcing was suggested, among others, by Belhajjame et al. [2] and McCann et al. [18]. In such lines of work, the main focus is on aggregating conflicting feedback. In an extended version of our empirical evaluation (not shown here due to space consideration) we have also shown the use of reasoning for correcting erroneous feedback. Belhajjame et al. [3] suggest the use of indirect feedback to improve the quality of a set of correspondences. We seek direct expert feedback, aiming at the minimization of the interaction.

Constraints were used before for schema mapping, *e.g.*, [11]. Our work focuses on the use of constraints for matching. Network level constraints, in particular the cycle constraints, were originally considered by Aberer et al. [1], [5], where they study the establishment of semantic interoperability in a large-scale P2P network. Probabilistic reasoning techniques are used for reconciliation without any user feedback.

Holistic matching, *e.g.*, [25] exploits the presence of multiple schemas, similar to our matching network, to improve the matching process. Nevertheless, this work aims at improving the initial matching outcome while our work uses the network to identify promising candidates for user feedback.

7 Conclusion and Future Work

In large-scale data integration scenarios, schema and ontology matching is complemented by a human-assisted post-matching reconciliation process. We analyzed this process in the setting of a matching network and introduced a formal model of matching reconciliation that uses human assertions over generic network-level constraints. Using the reasoning capabilities of ASP and simple yet generic constraints, as well as a heuristic ordering of the issues a human has to resolve, we were able to reduce the necessary user interactions by up to 40% compared to the baseline.

In future work, we aim at refining our notion of network-level constraints, which may include functional dependencies, foreign-key constraints and even domain-specific constraints that arise from a specific use case. A different, yet pragmatic direction, is to enforce constraints at a level finer than complete schemas, as schemas are often composed of meaningful building blocks.

Acknowledgements. This research has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 256955 and 257641.

References

1. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: Start making sense: The Chatty Web approach for global semantic agreements. *Journal of Web Semantics* 1(1), 89–114 (2003)
2. Belhajjame, K., Paton, N., Fernandes, A.A.A., Hedeler, C., Embury, S.: User feedback as a first class citizen in information integration systems. In: CIDR, pp. 175–183 (2011)
3. Belhajjame, K., Paton, N.W., Embury, S.M., Fernandes, A.A., Hedeler, C.: Incrementally improving dataspaces based on user feedback. *Information Systems* 38(5), 656–687 (2013)
4. Bernstein, P.A., Madhavan, J., Rahm, E.: Generic Schema Matching, Ten Years Later. *PVLDB* 4(11), 695–701 (2011)
5. Cudré-Mauroux, P., Aberer, K., Feher, A.: Probabilistic Message Passing in Peer Data Management Systems. In: ICDE, p. 41 (2006)
6. Do, H., Melnik, S., Rahm, E.: Comparison of schema matching evaluations. In: Proceedings of the 2nd Int. Workshop on Web Databases (German Informatics Society) (2002)
7. Do, H.H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches. In: VLDB, pp. 610–621 (2002)
8. Duchateau, F., Bellahsene, Z., Coletta, R.: Matching and Alignment: What Is the Cost of User Post-Match Effort? In: Meersman, R., Dillon, T., Herrero, P., Kumar, A., Reichert, M., Qing, L., Ooi, B.-C., Damiani, E., Schmidt, D.C., White, J., Hauswirth, M., Hitzler, P., Mohania, M. (eds.) OTM 2011, Part I. LNCS, vol. 7044, pp. 421–428. Springer, Heidelberg (2011)
9. Eiter, T., Ianni, G., Krennwallner, T.: Answer set programming: A primer. In: Tessaris, S., Franconi, E., Eiter, T., Gutierrez, C., Handschuh, S., Rousset, M.-C., Schmidt, R.A. (eds.) Reasoning Web. LNCS, vol. 5689, pp. 40–110. Springer, Heidelberg (2009)
10. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg, DE (2007)
11. Fagin, R., Haas, L.M., Hernández, M., Miller, R.J., Popa, L., Velegrakis, Y.: Clio: Schema mapping creation and data exchange. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 198–236. Springer, Heidelberg (2009)
12. Gal, A.: Uncertain Schema Matching. Morgan & Calypool Publishers (2011)
13. Gal, A., Sagi, T., Weidlich, M., Levy, E., Shafran, V., Miklós, Z., Hung, N.: Making sense of top-k matchings: A unified match graph for schema matching. In: Proceedings of SIGMOD Workshop on Information Integration on the Web, IIWeb 2012 (2012)
14. Galhardas, H., Lopes, A., Santos, E.: Support for user involvement in data cleaning. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2011. LNCS, vol. 6862, pp. 136–151. Springer, Heidelberg (2011)
15. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: ICLP/SLP, pp. 1070–1080. MIT Press (1988)
16. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. *Journal of New Generation Computing* 9(3/4), 365–386 (1991)
17. Jeffery, S.R., Franklin, M.J., Halevy, A.Y.: Pay-as-you-go user feedback for dataspace systems. In: SIGMOD, pp. 847–860 (2008)
18. McCann, R., Shen, W., Doan, A.: Matching Schemas in Online Communities: A Web 2.0 Approach. In: ICDE, pp. 110–119 (2008)
19. Peukert, E., Eberius, J., Rahm, E.: AMC - A framework for modelling and comparing matching systems as matching processes. In: ICDE, pp. 1304–1307 (2011)

20. Qi, Y., Candan, K.S., Sapino, M.L.: Ficsr: feedback-based inconsistency resolution and query processing on misaligned data sources. In: SIGMOD, pp. 151–162 (2007)
21. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal* 10(4), 334–350 (2001)
22. Reiter, R.: A logic for default reasoning. *Artificial Intelligence* 13(1-2), 81–132 (1980)
23. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering* (2012)
24. Smith, K.P., Morse, M., Mork, P., Li, M., Rosenthal, A., Allen, D., Seligman, L., Wolf, C.: The role of schema matching in large enterprises. In: CIDR (2009)
25. Su, W., Wang, J., Lochovsky, F.: Holistic schema matching for web query interfaces. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) *EDBT 2006. LNCS*, vol. 3896, pp. 77–94. Springer, Heidelberg (2006)
26. Yakout, M., Elmagarmid, A.K., Neville, J., Ouzzani, M., Ilyas, I.F.: Guided data repair. *Proc. VLDB Endow.* 4(5), 279–289 (2011)

Effective Recognition and Visualization of Semantic Requirements by Perfect SQL Samples

Van Bao Tran Le¹, Sebastian Link², and Flavio Ferrarotti¹

¹ School of Information Management,
Victoria University of Wellington, New Zealand

² Department of Computer Science,
University of Auckland, Auckland, New Zealand

{van.t.le,flavio.ferrarotti}@vuw.ac.nz, s.link@auckland.ac.nz

Abstract. SQL designs result from methodologies such as UML or Entity-Relationship models, description logics, or relational normalization. Independently of the methodology, sample data is promoted by academia and industry to visualize and consolidate the designs produced. SQL table definitions are a standard-compliant encoding of their designers' perception about the semantics of an application domain. Armstrong sample data visualize these perceptions. We present a tool that computes Armstrong samples for different classes of SQL constraints. Exploiting our tool, we then investigate empirically how these Armstrong samples help design teams recognize domain semantics. New measures empower us to compute the distance between constraint sets in order to evaluate the usefulness of our tool. Extensive experiments confirm that users of our tool are likely to recognize domain semantics they would overlook otherwise. The tool thereby effectively complements existing design methodologies in finding quality schemata that process data efficiently.

Keywords: Armstrong database, Empirical measure, Functional dependency, SQL, Uniqueness constraint.

1 Introduction

Design methodologies such as UML and Entity-relationship models, description logics, and relational normalization all have the ultimate goal to derive a quality database schema on which most frequent queries and most frequent updates can be processed efficiently. The output that these methodologies produce are usually relational database schemata. In practice, however, relational designs must be transformed into a standard-compliant SQL table format. While SQL is founded on the relational model of data, there is a strong disparity between them. For example, SQL permits occurrences of null markers and duplicate tuples to ease and speed-up data acquisition and processing. It is therefore a challenging task for design teams to find a suitable SQL table implementation. In particular, one that captures not only the structure but also the semantics of the application domain. Shortcomings in the acquisition of domain semantics are known to be

very costly, and often require expensive data cleaning and database repair techniques once the database is in use. These are some of the reasons why there is great consensus among academia and the commercial world that the use of good data samples can greatly benefit a more complete acquisition of requirements, and also help with the validation and consolidation of schemata produced automatically by design tools, as well as deliver a justification of the designs that can be effectively communicated to different stake-holders of the database.

In essence, Armstrong samples are data samples that satisfy the constraints a design team currently perceives as semantically meaningful for the application domain, and violate all those constraints that they currently perceive meaningless. Armstrong samples are therefore visualizations of abstract sets of constraints that encode real-world semantics. Intuitively, humans can learn a lot from Armstrong samples. They therefore hold great promise for the data-driven discovery of real-world application semantics. We illustrate this promise by revisiting a classical example.

Suppose our design team has arrived at the relation schema `CONTACT` with attributes *Street*, *City*, and *ZIP*, and the set Σ of the following functional dependencies (FDs): $Street, City \rightarrow ZIP$ and $ZIP \rightarrow City$. This is the classical example of a schema that is in Third normal form, but not in Boyce-Codd normal form. Normalization algorithms stop here, and cannot provide any further guidance on how to implement the relation schema within an SQL table definition. Magically, we may now produce an Armstrong *table* for the given set Σ of FDs and NOT NULL constraints, say on *Street* and *ZIP*, e.g. the table on the left:

<i>Street</i>	<i>City</i>	<i>ZIP</i>	<i>Street</i>	<i>City</i>	<i>ZIP</i>
03 Hudson St	Manhattan	10001	03 Hudson St	Manhattan	10001
03 Hudson St	Manhattan	10001	70 King St	Manhattan	10001
70 King St	Manhattan	10001	70 King St	San Fran	94107
70 King St	San Fran	94107	35 Lincoln Blvd	San Fran	94129
15 Maxwell St	San Fran	94129	15 Maxwell St	ni	60609
46 State St	ni	60609	15 Maxwell St	ni	60609

An inspection of this table shows that our specification of FDs does not exclude occurrences of duplicate tuples. More precisely, Σ does not imply any uniqueness constraints (UCs) over SQL tables. At this stage, our design team decides that the FD $Street, City \rightarrow ZIP$ should be replaced by the stronger UC $u(Street, City)$, meaning that there cannot be any different rows with matching total values on both *Street* and *City*. Note the interpretation of the null marker *ni* as *no information*, i.e., a value may not exist, or it may exist but is currently unknown. This is the interpretation that SQL uses [22]. The occurrence of *ni* in the table above indicates that the column *City* is nullable. An Armstrong table for the revised constraint set is shown on the right above. Looking at the last two rows of this table, the design team notices that the UC $u(Street, ZIP)$ is still not implied by the constraints specified so far. As the UC is considered to be meaningful, the designers decide to specify this constraint as well. Thus, the design team finally arrives at the following SQL table implementation

<pre>CREATE TABLE CONTACT (Street VARCHAR, City VARCHAR, ZIP INT, UNIQUE(Street,City), PRIMARY KEY(Street,ZIP), CHECK(Q = 0);</pre>	<pre>SELECT COUNT(*) FROM CONTACT c1 WHERE c1.ZIP IN (SELECT ZIP FROM CONTACT c2 WHERE c1.ZIP=c2.ZIP AND (c1.City <> c2.City OR (c1.City IS NULL AND c2.City IS NOT NULL) OR (c1.City IS NOT NULL AND c2.City IS NULL)));</pre>	<i>Q</i>
--	--	----------

where the state assertion *Q* on the right enforces $ZIP \rightarrow City$.

Contributions. As our main contribution we investigate empirically how Armstrong samples in standard-compliant SQL format help design teams recognize domain semantics. For this contribution we need to overcome several obstacles. In previous research, Armstrong samples have been investigated in the context of the relational model of data only. Thus, Armstrong relations cannot show the delicate interactions between SQL constraints. However, we draw from our recent work where we developed algorithms to compute Armstrong tables for different classes of SQL constraints. Our first contribution in this paper is a tool that is the first to implement these algorithms. We describe the graphical user interface of the tool and its functionality. Our second contribution is the definition of several empirical measures to assess the effectiveness of using our tool. In essence, *soundness* measures how many of the as meaningful perceived constraints are actually meaningful, *completeness* measures how many of the actually meaningful constraints are perceived as meaningful, and *proximity* combines soundness and completeness. Our measures assess the quality of a constraint set with respect to a target constraint set, and therefore qualify naturally for the use in automated assessment tools, e.g., in database courses. In the example above, the target set Σ_t consists of the UCs $u(Street, City)$, $u(Street, ZIP)$, and the FD $ZIP \rightarrow City$. The original set Σ of FDs consisting of $Street, City \rightarrow ZIP$ and $ZIP \rightarrow City$ is fully sound with respect to Σ_t since both FDs in Σ are implied by Σ_t . While Σ is also fully complete with respect to FDs, it is fully incomplete with respect to UCs. That is, every FD implied by Σ_t is also implied by Σ , but no UC implied by Σ_t is implied by Σ . In our third contribution we present an analysis of our extensive experiments with our tool. Our experiments determine what and how much design teams learn about the application domain in addition to what they know prior to inspecting Armstrong tables produced by our tool. Our analysis shows that inspecting Armstrong tables has no impact on recognizing meaningless SQL constraints which are incorrectly perceived as meaningful, but inspecting Armstrong tables empowers design teams to recognize nearly all meaningful SQL constraints that are incorrectly perceived as meaningless. These results empirically confirm our intuition that *the satisfaction of meaningless SQL constraints is nearly impossible to be observed, and the violation of meaningful SQL constraints is almost certain to be observed in Armstrong tables*.

Organization. We summarize related work in Section 2 and define the necessary framework in Section 3. Our tool is presented in Section 4, and an overview of the

experimental design is given in Section 5. Our measures are defined in Section 6, and a quantitative and qualitative analysis is presented in Section 7. We conclude in Section 8 where we also briefly comment on future work.

2 Related Work

Armstrong databases have been regarded intuitively as a conceptual tool helpful with the acquisition of domain semantics [5, 17–20]. Theoretical work on computational and structural properties of Armstrong databases in the relational model of data and the Entity-Relationship model are manifold, including [2, 3, 6, 12, 18, 21] with survey papers [5, 17].

One of the most important extensions of Codd’s basic relational model [4] is partial information to cope with the high demand for the correct handling of such information in real-world applications. In the literature many interpretations of null markers have been proposed such as “missing” or “value unknown at present”, “no information”, and “inapplicable”. Our tool can handle constraints under the two most popular interpretations: null marker occurrences interpreted as “value unknown at present” are denoted by *unk*, and occurrences that are interpreted as “no information” are denoted by *ni*. As the latter is the one used by SQL, our presentation in this paper will mostly focus on the “no information” interpretation.

In recent research we have established a theory of Armstrong tables for different classes of NOT NULL constraints, uniqueness constraints (UCs) and functional dependencies (FDs) under occurrences of either the *unk* marker, or the *ni* marker [7, 9–11, 14]. The present article is the first to present a tool for the computation of Armstrong tables under different classes of constraints and the two different null marker interpretations. The tool subsumes those implemented for relational databases as an idealized special case [18, 19].

Our tool forms the critical basis for our empirical investigations into the usefulness of Armstrong tables. We utilize the tool to compute Armstrong tables on-the-fly and in response to inputs by design teams. The teams then inspect the Armstrong tables together with domain experts in order to consolidate their perception of the domain semantics in form of a set of SQL constraints. For the class of FDs over strictly relational databases our previous research has shown that the use of Armstrong relations is likely to increase the recognition of meaningful FDs, but unlikely to increase the recognition of meaningless FDs [13]. Since the interaction of SQL constraints is more involved than that for their idealized relational counterparts, one would naturally assume that sample data becomes even more useful. In the current paper we therefore extend the experimental framework from [13] to investigate the usefulness, and exploit our tool in actual experiments. Our new measures do not just address the single class of FDs, but the two classes of UCs and FDs. This is necessary as NOT NULL constraints, UCs and FDs interact non-trivially over SQL data, in contrast to relational data where all attributes are NOT NULL and UCs are simply subsumed by FDs. In addition to the measures of *soundness*, *completeness* and *proximity*,

we introduce here their *relative versions*. These illustrate best our findings: almost all UCs and FDs that can possibly be recognized are actually recognized by inspecting Armstrong tables, and that Armstrong tables do not have an impact on recognizing meaningless UCs and FDs incorrectly perceived as meaningful.

3 Preliminaries

In this section we define the syntax and semantics for the different classes of constraints under different interpretations of null markers. While an exact understanding of the semantics is not necessary to appreciate our main results, they are required to fully appreciate the features implemented in our tool.

Let $\mathfrak{H} = \{H_1, H_2, \dots\}$ be a countably infinite set of symbols, called (*column*) *headers*. A *table schema* is a finite non-empty subset T of \mathfrak{H} . Each header H of a table schema T is associated with an infinite domain $\text{dom}(H)$ of the possible values that can occur in column H . To encompass partial information every column may contain occurrences of a null marker, $\text{ni} \in \text{dom}(H)$.

For header sets X and Y we may write XY for $X \cup Y$. If $X = \{H_1, \dots, H_m\}$, then we may write $H_1 \cdots H_m$ for X . In particular, we may write H to represent $\{H\}$. A *row* over T is a function $r : T \rightarrow \bigcup_{H \in T} \text{dom}(H)$ with $r(H) \in \text{dom}(H)$ for all $H \in T$. For $X \subseteq T$ let $r(X)$ denote the restriction of the row r over T to X . An *SQL table* t over T is a finite multi-set of rows over T . For rows r_1 and r_2 over T , r_1 *subsumes* r_2 if for all $H \in T$, $r_1(H) = r_2(H)$ or $r_2(H) = \text{ni}$.

For a row r over T and a set $X \subseteq T$, r is said to be X -total if for all $H \in X$, $r(H) \neq \text{ni}$. Similar, an SQL table t over T is said to be X -total, if every row r of t is X -total. An SQL table t over T is said to be *total* if it is T -total.

A *null-free subschema* (NFS) over the table schema T is an expression $nfs(T_s)$ where $T_s \subseteq T$. The NFS $nfs(T_s)$ over T is satisfied by an SQL table t over T , denoted by $\models_t nfs(T_s)$, if and only if t is T_s -total. In practice, the NFS consists of those attributes declared NOT NULL in the SQL table definition.

An *SQL functional dependency* (SFD) over a table schema T is an expression $X \rightarrow Y$ where $X, Y \subseteq T$. An SQL table t over T satisfies the SFD $X \rightarrow Y$ if for all rows $r, r' \in t$ the following holds: if $r(X) = r'(X)$ and r, r' are both X -total, then $r(Y) = r'(Y)$ [16]. An *SQL uniqueness constraint* (SUC) over table schema T is an expression $u(X)$ where $X \subseteq T$. An SQL table t satisfies the SUC $u(X)$ if for all rows $r, r' \in t$ the following holds: if $r(X) = r'(X)$ and both r and r' are X -total, then $r = r'$. For examples, both SQL tables from the introduction satisfy $ZIP \rightarrow City$. While the left table violates every SUC, the right table satisfies $u(Street, City)$ but violates $u(Street, ZIP)$.

Let \mathcal{C} be a class of constraints, for example, the combined class of NOT NULL constraints, SUCs and SFDs. We say for a set $\Sigma \cup \{\varphi\}$ of constraints from \mathcal{C} over table schema T that Σ *implies* φ , denoted by $\Sigma \models \varphi$, if for every SQL table t over T that satisfies every constraint in Σ , t also satisfies φ . For example, the right SQL table from the introduction shows that the set Σ consisting of $ZIP \rightarrow City$, $u(Street, City)$, and the NFS $nfs(Street, ZIP)$ does not imply $u(Street, ZIP)$.

For a set Σ of constraints in \mathcal{C} over table schema T , we say that an SQL table t over T is \mathcal{C} -Armstrong for Σ if t satisfies every constraint in Σ , and violates every constraint in \mathcal{C} over T that is not implied by Σ . For example, the table on the right from the introduction is Armstrong for the set Σ containing $ZIP \rightarrow City$, $u(Street, City)$, and $nfs(Street, ZIP)$. By inspecting this table, we know that Σ does not imply $City \rightarrow Street$ nor $u(Street, ZIP)$, but does imply $Street, ZIP \rightarrow City$ and $u(Street, City)$.

For our experiments we will focus on SQL constraints exclusively. Constraints, however, can also be defined on tables that feature the Codd null marker `unk`, instead of `ni`. In that case we speak of *Codd tables*. For a Codd table t over T , the set $Poss(t)$ of all possible worlds relative to t is defined by $Poss(t) = \{t' \mid t'$ is a table over T and there is a bijection $b : t \rightarrow t'$ such that $\forall r \in t, r$ is subsumed by $b(r)$ and $b(r)$ is T -total}. A *Codd functional dependency* (CFD) over table schema T is an expression $\diamond(X \rightarrow Y)$ where $X, Y \subseteq T$. A Codd table t over T satisfies $\diamond(X \rightarrow Y)$ if there is some $p \in Poss(t)$ such that for all rows $r, r' \in p$ the following holds: if $r(X) = r'(X)$, then $r(Y) = r'(Y)$. A *Codd uniqueness constraint* (CUC) over table schema T is an expression $\diamond u(X)$ where $X \subseteq T$. A Codd table t satisfies $\diamond u(X)$ if there is some $p \in Poss(t)$ such that for all rows $r, r' \in p$ the following holds: if $r(X) = r'(X)$ and both r and r' are X -total, then $r = r'$. The notions of implication and Armstrong tables, defined in the context of SQL tables above, are defined analogously in the context of Codd tables.

Algorithms to compute \mathcal{C} -Armstrong tables were recently developed for the classes \mathcal{C} of NOT NULL constraints and i) SUCs in [10], ii) SUCs and SFDs in [9], iii) CUCs in [14], and iv) CUCs and CFDs in [7]. Our tool implements all of these algorithms, but for the experiments we focus on the class ii) above.

4 A Tool to Recognize and Visualize Domain Semantics

In this section we present the functionality of our tool. It is the first to implement recently published algorithms for computing Armstrong tables [7, 9, 10, 14]. It is a Web application, developed in the .NET framework, and operates on common Internet browsers such as Firefox, Google Chrome, and Internet Explorer. We invite the reader to visit <http://armstrongtable.sim.vuw.ac.nz> to experiment with the tool. A screenshot of the GUI is shown in Figure 1.

The workflow of the tool exploits the following components. Firstly, the user selects a context which fixes the interpretation of occurring null markers and the class of constraints. If `ni` is selected, the user chooses from the class of SQL UCs, or the combined class of SQL UCs and SQL FDs. Otherwise, the user chooses from the class of Codd UCs, or the combined class of Codd UCs and Codd FDs.

Secondly, the user specifies i) the table schema and which column headers are declared NOT NULL, ii) the sets of constraints for the classes of constraints from the context, and iii) optionally, some designated domain values to populate the Armstrong table.

Thirdly, the user choose from several algorithms. Foremost, a \mathcal{C} -Armstrong table can be computed for the set Σ of input constraints from \mathcal{C} . The tool populates the table by either domain values provided or by artificial values. The user can replace values in the table by new values such that the new table is also \mathcal{C} -Armstrong for Σ . Depending on the class of constraints selected, other algorithms include the computation of closures for a collection of column headers, the set of anti-keys, the set of maximal sets for each column header, and duplicate sets. These notions are defined in our previous work [7, 9, 10, 14].

Finally, users of the tool have the option to export the table to an XML file, or print it out. The XML file may be used in many other common applications, including Microsoft Office, OpenOffice, or Apple's iWork.

The screenshot shows a web-based application titled "ARMSTRONG TABLE COMPUTATION". The top navigation bar includes links for "Armstrong Table", "Schema Input", "Computation Results", "User Account", and "About". On the right, there is a "Welcome Sebo Link" and a "Log Out" button. Below the navigation is a toolbar with icons for various file formats: Word, Excel, PDF, and others. The main content area is titled "ARMSTRONG TABLE COMPUTATION RESULT SETS". It features a table with columns "Street", "City", and "ZIP". The data rows are:

Street	City	ZIP
03 Hudson St	Manhattan	10001
70 King St	Manhattan	10001
35 Lincoln Blvd	San Fran	94129
35 Lincoln Blvd	Los Angeles	90045
Street5	City5	ZIP5
Street6	City5	ZIP6
Street7	ni	ZIP7
Street7	ni	ZIP7

Below the table, two formulas are displayed: FDs(R) = { ZIP->City } and UCs(R) = { Street,City }. A "Customise Armstrong Table Values" button is located at the bottom of the table area.

Fig. 1. GUI of web-based tool: Armstrong table with some customized domain values

5 Experimental Design

Our aim is to evaluate how well Armstrong tables help design teams recognize domain semantics. We thus ask how much domain semantics they learn by inspecting Armstrong tables *in addition* to what they can learn without them.

Overview. Naturally, our experiment consists of two phases. In the first phase, each design team i is given the same application domain in form of a table schema and NOT NULL constraints. A natural language description accompanies the schema definition, and domain experts are available to answer questions about the domain. The experts have no database background and do not answer questions about database constraints. After internal discussion and consultation with the experts, each team i writes down a cover Σ_1^i of SQL UCs and FDs that they perceive as meaningful for the domain. For each team i , our tool computes an Armstrong table for Σ_1^i and the NOT NULL constraints.

In the second phase, each team i revises their constraint set Σ_1^i with the help of the Armstrong table for Σ_1^i . Again, domain experts can be consulted and the Armstrong table can be used to communicate with them. Teams are provided with Armstrong tables for their revised constraint sets until they are happy with the outcome. The final constraint set is denoted by Σ_2^i .

For each team i , we use different measures (see the next section) to compare Σ_1^i with our target set Σ_t of constraints, and to compare Σ_2^i with Σ_t . If the latter comparison results in a higher similarity than the former comparison, we conclude that Armstrong tables are indeed useful with respect to the measure. The comparison between Σ_1^i and Σ_t gives us a baseline of what team i can still possibly learn by inspecting Armstrong tables. Similarly, the comparison between Σ_2^i and Σ_t tells us how much team i has actually learned by inspecting Armstrong tables. Graphs will illustrate for each team the difference between how much they could possibly learn and how much they have actually learned.

Design Teams and Domain Experts. The experiments involved 50 design teams from three universities: the University of Auckland, the Victoria University of Wellington, and the Lotus University of Vietnam. The students took third year database courses, were familiar with the semantics of SQL constraints, and that the Armstrong tables satisfy the constraints they currently perceive as meaningful and violate every constraint they currently perceive as meaningless. Each team consisted of 2 or 3 students.

The authors of this paper acted as domain experts for the given application domain. They were present during the experiments to clarify questions by the teams. Teams were not given advice about the specification of their constraint sets. For example, questions like "Does this UC make sense?" were not answered. In practice, there may not exist a unique target set, because the available information may not identify a unique constraint set. For conducting the experiment, however, the presence of the domain experts and their consensus on the target set guarantees that the quality of constraint sets can be measured transparently.

Application Domain and Target. As application domain we used the schema $WORK=\{P(roj),E(mp),D(ate),R(ole),H(rs)\}$ with the following description. The schema records information about the number Hrs of hours (e.g., 5) that an employee Emp (e.g., Dilbert) works on a project $Proj$ (e.g., *Blue*) in a role $Role$ (e.g., *Programmer*) at a day $Date$ (e.g., *Oct 5*). The null-free subschema of $WORK$ is $nfs(Emp, Date)$, i.e., ni must not occur in the Emp and $Date$ columns.

Σ_t contains $nfs(Emp, Date)$ and the following SQL UCs and FDs. The UC $u(Emp, Date)$ states that there cannot be different rows with the same employee and the same date. The FD $Proj, Emp \rightarrow Role$ says that in each project, each employee has a unique role. The FD $Project, Role \rightarrow Hrs$ says that the project and role together determine the number of hours. Finally, the FD $Proj, Emp \rightarrow Hrs$ says that the project and employee together determine the number of hours. The last FD is not implied by the other UCs and FDs in Σ_t [11].

Limitations. These include issues such as students acting as database designers, the familiarity of students with the application domain, the number and size of

the application domain, time constraints, the assumption that domain experts are present, and that there is consensus among them. The discussion of these limitations from the idealized relational case [13] is also valid for SQL constraints.

6 Quality Measures

We define new measures to compare constraint sets, and illustrate these on the following running example from our actual experiment. Design team 8 handed in the following sets of constraints:

- $\Sigma_1^8 = \{u(Emp, Proj, Date); Proj, Role \rightarrow Hrs\}$
- $\Sigma_2^8 = \{u(Emp, Date); Proj, Role \rightarrow Hrs; Proj, Emp \rightarrow Role\}$

Our measures will enable us to compare these sets to the target set Σ_t . As the NFS $nfs(Emp, Date)$ is given, it belongs to all sets Σ_j^i , $i = 1, \dots, 50$ and $j = 1, 2$.

Soundness measures which of the constraints perceived meaningful by a team, are actually meaningful. Here, actually meaningful are the constraints implied by the target set Σ_t . The UCs implied by a set Σ of UCs and FDs and $nfs(T_s)$ is defined as $s(\Sigma) := \{u(X) \mid \Sigma \models u(X)\}$. Soundness for UCs is thus the ratio between the as meaningful perceived UCs that are implied by Σ_t and all the as meaningful perceived UCs: $sound_{\Sigma_t}^u(\Sigma) = |s(\Sigma) \cap s(\Sigma_t)| / |s(\Sigma)|$, and $sound_{\Sigma_t}^u(\Sigma) := 1$, if $s(\Sigma) = \emptyset$. For the measures of FDs we exploit the notion of a closure $X_\Sigma^* = \{H \in T \mid \Sigma \models X \rightarrow H\}$ for a set X of headers under Σ . Let $\mathcal{P}_0(T)$ denote the set of all non-empty, proper subsets of T . Then the soundness for FDs is the ratio between the header sets in $\mathcal{P}_0(T)$ whose closure under Σ is contained in the closure under Σ_t , and $\mathcal{P}_0(T)$: $sound_{\Sigma_t}^f(\Sigma) = |\{X \in \mathcal{P}_0(T) \mid X_\Sigma^* \subseteq X_{\Sigma_t}^*\}| / |\mathcal{P}_0(T)|$. For our running example we obtain $sound_{\Sigma_t}^u(\Sigma_1^8) = 1$, $sound_{\Sigma_t}^f(\Sigma_1^8) = 30/30 = 1$, and $sound_{\Sigma_t}^u(\Sigma_2^8) = 8/8 = 1$, $sound_{\Sigma_t}^f(\Sigma_2^8) = 1$.

Completeness measures which of the actually meaningful constraints are also perceived as meaningful by a team. Completeness for UCs is thus the ratio between the as meaningful perceived UCs that are implied by Σ_t and all the actually meaningful UCs: $complete_{\Sigma_t}^u(\Sigma) = |s(\Sigma) \cap s(\Sigma_t)| / |s(\Sigma_t)|$, and $complete_{\Sigma_t}^u(\Sigma) := 1$, if $s(\Sigma_t) = \emptyset$. Completeness for FDs is the ratio between the header sets in $\mathcal{P}_0(T)$ whose closure under Σ_t is contained in the closure under Σ , and $\mathcal{P}_0(T)$: $complete_{\Sigma_t}^f(\Sigma) = |\{X \in \mathcal{P}_0(T) \mid X_{\Sigma_t}^* \subseteq X_\Sigma^*\}| / |\mathcal{P}_0(T)|$. For our running example, we obtain $complete_{\Sigma_t}^u(\Sigma_1^8) = 4/8 = 0.5$, $complete_{\Sigma_t}^f(\Sigma_1^8) = 26/30 = 0.8$, and $complete_{\Sigma_t}^u(\Sigma_2^8) = 8/8 = 1$, $complete_{\Sigma_t}^f(\Sigma_2^8) = 29/30 \approx 0.97$.

Proximity measures how close two sets of constraints are. For UCs it is the ratio between the as meaningful perceived UCs that are implied by Σ_t and all the actually meaningful and all the as meaningful perceived UCs: $prox^u(\Sigma, \Sigma_t) = |s(\Sigma) \cap s(\Sigma_t)| / |s(\Sigma) \cup s(\Sigma_t)|$, and $prox^u(\Sigma, \Sigma_t) := 1$, if $s(\Sigma_t) \cup s(\Sigma_t) = \emptyset$. The complement $dist^u(\Sigma, \Sigma_t) = |(s(\Sigma) \cup s(\Sigma_t)) - (s(\Sigma) \cap s(\Sigma_t))|$ defines a metric on equivalent sets of UCs. For FDs, completeness is the ratio between the header sets in $\mathcal{P}_0(T)$ whose closure under Σ_t is the same as the closure under Σ , and $\mathcal{P}_0(T)$: $prox^f(\Sigma, \Sigma_t) = |\{X \in \mathcal{P}_0(T) \mid X_{\Sigma_t}^* = X_\Sigma^*\}| / |\mathcal{P}_0(T)|$. Similar to UCs,

$dist^f(\Sigma, \Sigma_t) = |\{X \in \mathcal{P}_0(T) \mid X_{\Sigma_t}^* \neq X_\Sigma^*\}|$ defines a metric on equivalent sets of FDs. For our example, we obtain $prox(\Sigma_1^8, \Sigma_t) = 4/8 = 0.5$, $prox^f(\Sigma_1^8, \Sigma_t) = 26/30 = 0.8$, and $prox^u(\Sigma_2^8, \Sigma_t) = 8/8 = 1$, $prox^f(\Sigma_2^8, \Sigma_t) = 29/30 \approx 0.97$.

For the best insight into the usefulness of Armstrong tables we present, for each of the measures, the difference between how much Armstrong tables can *possibly* and *actually* improve the measurement. In what follows we refer by $measure(\Sigma)$ to one of $sound^u_{\Sigma_t}(\Sigma)$, $sound^f_{\Sigma_t}(\Sigma)$, $complete^u_{\Sigma_t}(\Sigma)$, $complete^f_{\Sigma_t}(\Sigma)$, $prox^u(\Sigma, \Sigma^t)$ and $prox^f(\Sigma, \Sigma^t)$. Then we define $possible-gain-measure^i := (1 - measure(\Sigma_1^i)) \cdot 100\%$, and $actual-gain-measure^i := (measure(\Sigma_2^i) - measure(\Sigma_1^i)) \cdot 100\%$. Note that actual gains can also be negative. For our running example we obtain $possible-gain-UC-sound^8 = 0\%$, $actual-gain-UC-sound^8 = 0\%$, $possible-gain-FD-sound^8 = 0\%$, $actual-gain-FD-sound^8 = 0\%$, $possible-gain-UC-complete^8 = 50\%$, $actual-gain-UC-complete^8 = 50\%$, $possible-gain-FD-complete^8 = 20\%$, and $actual-gain-FD-complete^8 = 17\%$.

7 Data Analysis

We analyze our experiments with 50 design teams quantitatively and qualitatively. Due to lack of space we will focus on our main findings and will not analyze proximity separately, as it combines soundness and completeness.

7.1 Quantitative Analysis

The following tables show some statistics for the actual gains teams achieved by inspecting Armstrong tables. Means are arithmetic means.

Gain-in-soundness in percent				
Class	Min	Mean	Median	Max
UCs	-33	4	0	44
FDs	-13	0	0	14

Gain-in-completeness in percent				
Class	Min	Mean	Median	Max
UCs	0	24	25	75
FDs	0	11	12	27

The statistics confirm our intuition that Armstrong tables are unlikely to help design teams recognize meaningless constraints that they perceive meaningful prior to inspecting a corresponding Armstrong table. While a small number of teams have added new meaningless constraints (a negative gain in soundness), a small number of different teams have removed meaningless constraints (a positive gain). On average, these gains and losses even each other out. The statistics also confirm our intuition that Armstrong tables are likely to help design teams recognize meaningful constraints they perceive meaningless prior to inspecting an Armstrong table. Indeed, no meaningful constraints are removed at all, but a significant number of meaningful constraints is added on average.

Figures 2 to 5 strengthen these observation. For soundness it shows that actual gains are similar to no gains. For completeness it shows that possible and actual gains nearly coincide.

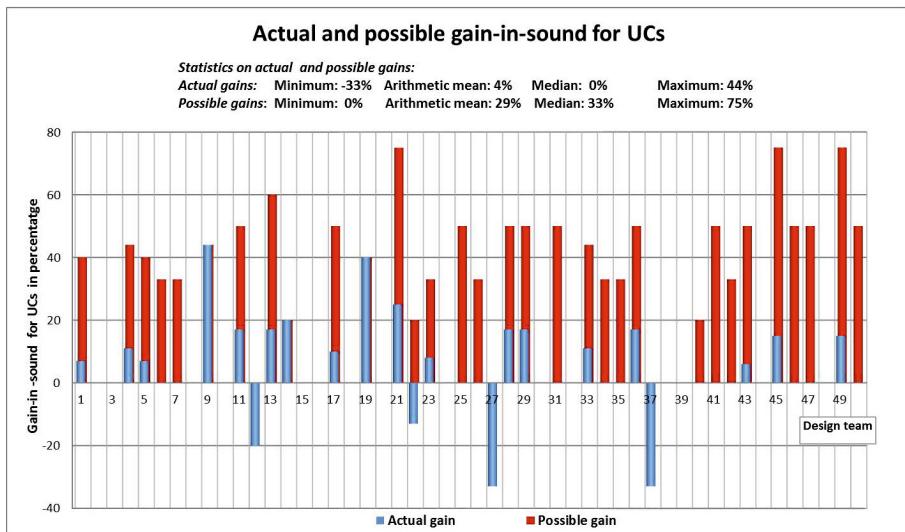


Fig. 2. Actual and Possible Gains for Soundness of Uniqueness Constraints

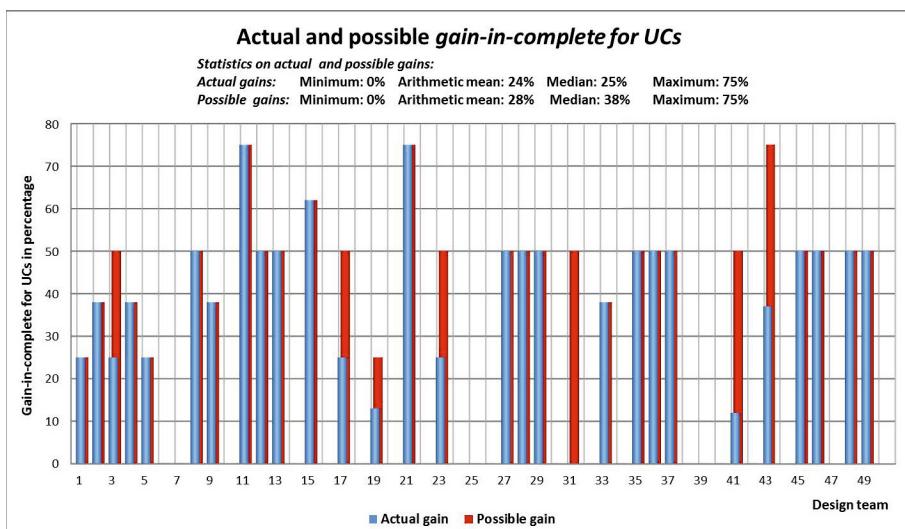
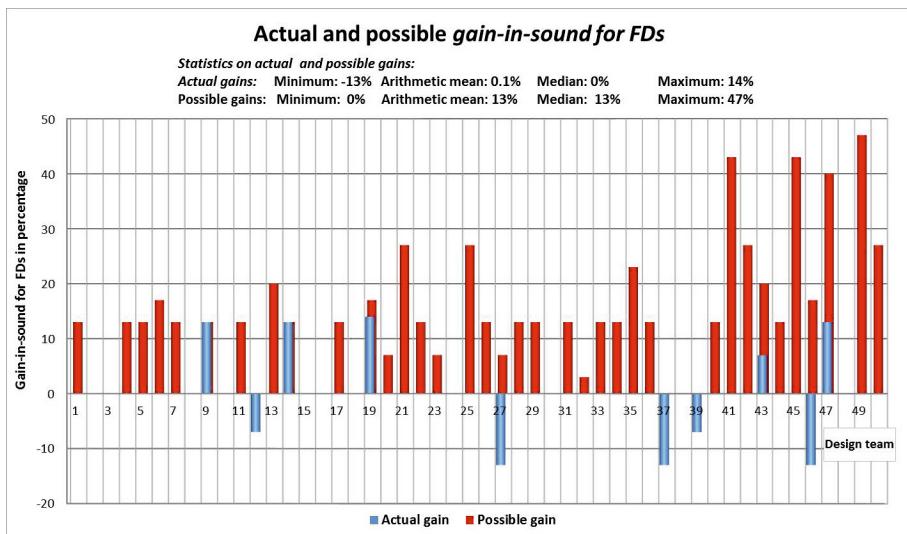
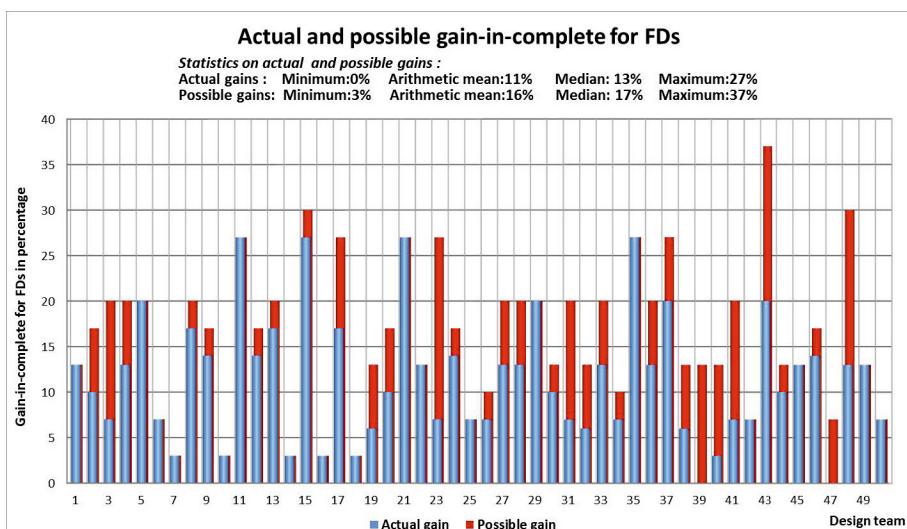


Fig. 3. Actual and Possible Gains for Completeness of Uniqueness Constraints

**Fig. 4.** Actual and Possible Gains for Soundness of Functional Dependencies**Fig. 5.** Actual and Possible Gains for Completeness of Functional Dependencies

7.2 Qualitative Analysis

A qualitative analysis of the soundness confirms that only few teams add or remove meaningless constraints after inspecting Armstrong tables. Indeed, three teams added $u(EP)$ and one team added $u(EPR)$, and $PE \rightarrow D$, $PD \rightarrow E$, and $ER \rightarrow P$ were each added by at most one team. Three teams removed $u(EP)$ and one team removed $u(DERH)$. $PR \rightarrow E$, $PR \rightarrow D$, $ER \rightarrow P$, $DH \rightarrow E$, $R \rightarrow P$ were each removed by at most one team.

A qualitative analysis of the completeness confirms that significant numbers of teams add meaningful constraints after inspecting Armstrong tables. Indeed, 22 teams added $u(ED)$, and 41 teams added some meaningful FD: 18 teams added $PR \rightarrow H$, 15 teams added $PE \rightarrow H$, and 14 teams added $EP \rightarrow R$. The numbers also suggest that there is no particular pattern on which meaningful FD is added. Importantly, no team removed any meaningful constraint.

8 Conclusion and Future Work

Conceptual methodologies produce relational approximations of database designs that still need to be converted into real-world SQL table definitions. In particular, the semantics of the application domain must be encoded in form of SQL constraints. We have investigated how much perfect SQL sample data, in form of Armstrong tables, help design teams recognize uniqueness constraints and functional dependencies that are meaningful for the underlying application domain. For this purpose we developed a tool that implements recent algorithms for the computation of Armstrong tables. The tool was then used in our experiments to create Armstrong tables for sets of these SQL constraints that design teams perceive as meaningful. New measures were exploited to confirm empirically that the inspection of Armstrong tables is likely to help design teams recognize nearly all meaningful SQL constraints they did not recognize before, but unlikely to help them recognize any actually meaningless SQL constraints they perceive as meaningful. The results extend previous findings for purely relational functional dependencies. They suggest to use our tool as early as possible during requirements acquisition and to exploit it for the consolidation and visualization of database designs produced by popular conceptual design methodologies.

In future work we will address the limitations of our experiments, include other classes of SQL constraints such as cardinality and referential constraints [8, 15, 21], gather data on how the size of Armstrong tables affects the recognition of domain semantics, implement our measures for the automated assessment and feedback of exercises in database courses, and combine our approach with techniques from natural language processing [1]. In particular, our results may not apply in this form when different constraints or constraints on several tables are considered, such as foreign keys.

Acknowledgement. This research is supported by the Marsden fund council from Government funding, administered by the Royal Society of New Zealand. We express our sincere gratitude to Pavle Mogin and Hui Ma for their kind assistance during the data gathering process.

References

1. Albrecht, M., Buchholz, E., Düsterhöft, A., Thalheim, B.: An informal and efficient approach for obtaining semantic constraints using sample data and natural language processing. In: Libkin, L., Thalheim, B. (eds.) Semantics in Databases 1995. LNCS, vol. 1358, pp. 1–28. Springer, Heidelberg (1998)
2. Beeri, C., Dowd, M., Fagin, R., Statman, R.: On the structure of Armstrong relations for functional dependencies. *J. ACM* 31(1), 30–46 (1984)
3. Beskales, G., Ilyas, I., Golab, L.: Sampling the repairs of functional dependency violations under hard constraints. *PVLDB* 3(1), 197–207 (2010)
4. Codd, E.F.: A relational model of data for large shared data banks. *Commun. ACM* 13(6), 377–387 (1970)
5. Fagin, R.: Armstrong databases. Tech. Rep. RJ3440(40926), IBM Research Laboratory, San Jose, California, USA (1982)
6. Fagin, R.: Horn clauses and database dependencies. *J. ACM* 29(4), 952–985 (1982)
7. Ferrarotti, F., Hartmann, S., Le, V., Link, S.: Codd table representations under weak possible world semantics. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part I. LNCS, vol. 6860, pp. 125–139. Springer, Heidelberg (2011)
8. Ferrarotti, F., Hartmann, S., Link, S.: Efficiency frontiers of XML cardinality constraints. *Data Knowl. Eng.* (2013), <http://dx.doi.org/10.1016/j.datak.2012.09.004>
9. Hartmann, S., Kirchberg, M., Link, S.: Design by example for SQL table definitions with functional dependencies. *VLDB J.* 21(1), 121–144 (2012)
10. Hartmann, S., Leck, U., Link, S.: On Codd families of keys over incomplete relations. *Comput. J.* 54(7), 1166–1180 (2011)
11. Hartmann, S., Link, S.: The implication problem of data dependencies over SQL table definitions. *ACM Trans. Database Syst.* 37(2), 13 (2012)
12. Hartmann, S., Link, S., Trinh, T.: Constraint acquisition for Entity-Relationship models. *Data Knowl. Eng.* 68(10), 1128–1155 (2009)
13. Langeveldt, W.D., Link, S.: Empirical evidence for the usefulness of Armstrong relations in the acquisition of meaningful functional dependencies. *Inf. Syst.* 35(3), 352–374 (2010)
14. Le, V., Link, S., Memari, M.: Schema- and data-driven discovery of SQL keys. *JCSE* 6(3), 193–206 (2012)
15. Liddle, S.W., Embley, D.W., Woodfield, S.N.: Cardinality constraints in semantic data models. *Data Knowl. Eng.* 11(3), 235–270 (1993)
16. Lien, E.: On the equivalence of database models. *J. ACM* 29(2), 333–362 (1982)
17. Link, S.: Armstrong databases: Validation, communication and consolidation of conceptual models with perfect test data. In: Ghose, A., Ferrarotti, F. (eds.) APCCM 2012, pp. 3–20. Australian Computer Society (2012)
18. Mannila, H., Räihä, K.J.: Design of Relational Databases. Addison-Wesley (1992)
19. Silva, A., Melkanoff, M.: A method for helping discover the dependencies of a relation. In: Advances in Data Base Theory, pp. 115–133 (1979)
20. Thalheim, B.: Entity-Relationship modeling. Springer (2000)
21. Thalheim, B.: Fundamentals of cardinality constraints. In: Pernul, G., Tjoa, A.M. (eds.) ER 1992. LNCS, vol. 645, pp. 7–23. Springer, Heidelberg (1992)
22. Zaniolo, C.: Database relations with null values. *J. Comput. Syst. Sci.* 28(1), 142–166 (1984)

A Semantic Approach to Keyword Search over Relational Databases

Zhong Zeng, Zhifeng Bao, Mong Li Lee, and Tok Wang Ling

School of Computing, National University of Singapore
`{zengzh,baozhife,leeml,lingtw}@comp.nus.edu.sg`

Abstract. Research in relational keyword search has been focused on the efficient computation of results as well as strategies to rank and output the most relevant ones. However, the challenge to retrieve the intended results remains. Existing relational keyword search techniques suffer from the problem of returning overwhelming number of results, many of which may not be useful. In this work, we adopt a semantic approach to relational keyword search via an *Object-Relationship-Mixed data graph*. This graph is constructed based on database schema constraints to capture the semantics of objects and relationships in the data. Each node in the ORM data graph represents either an object, or a relationship, or both. We design an algorithm that utilizes the ORM data graph to process keyword queries. Experiment results show our approach returns more informative results compared to existing methods, and is efficient.

Keywords: Keyword Search, Relational Databases, Semantic Approach.

1 Introduction

The success of web search engines has made keyword search the most popular search paradigm for ordinary users. Given the rapid growth of structured data repositories, the ability to support keyword search over such repositories enables users to pose keyword queries easily without the need to have full knowledge of the database schemas or structured query languages.

Research in relational keyword search has been focused on the efficiency of computation of results from multiple tuples [9,12,8,6,4] as well as ranking strategies to improve the quality of results [16,17,20]. The works in [16,17,12,14,13] examine the effectiveness of relational keyword queries. However, the retrieval of informative results for relational keyword search remains a challenge.

We observe that when a user issues a keyword query, each keyword is usually directed at some object of interest, or relationship along with the associated objects. This motivates us to design a semantic approach to increase the effectiveness of relational keyword queries. In particular, we will construct an *Object-Relationship-Mixed data graph* (ORM data graph) of the database which consists of three types of nodes, namely object node, relationship node and mixed type node. In contrast to the traditional data graph where each node corresponds to a

Student				Qualification				
tupleid	sid	name	sex	tupleid	lid	degree	major	university
s1	U054	John Williams	Male	q1	StnL	PhD	CS	University of Wisconsin-Madison
s2	A005	Edward Martin	Male	q2	StnL	Master	EE	University of Toronto
s3	A021	Mary Smith	Female	q3	JntK	PhD	CS	National University of Singapore

Course					Enrol			
tupleid	cid	title	credit	lid	tupleid	sid	cid	grade
c1	CS421	Database Design	4.0	StnL	e1	U054	CS203	A
c2	CS526	Information Retrieval	3.0	JntK	e2	U054	CS421	B
c3	CS203	Java Programming	3.5	JntK	e3	A005	CS421	A

Lecturer				
tupleid	lid	Name	office	email
l1	StnL	Steven Lee	COM2 215	slee@yyy.zz
l2	JntK	Janet Kate	COM1 316	jkate@mnn.mn

Fig. 1. Example relational database

tuple, a node in an ORM data graph may correspond to a list of tuples. We will show that the ORM data graph can facilitate the retrieval of useful and relevant information for relational keyword queries.

The contributions of our work are summarized as follows:

1. We identify limitations of existing approaches for relational keyword search as they do not consider the objects and relationships represented by data instances.
2. We design an ORM data graph of the database to capture the semantics of objects and relationships in the data. Based on this graph, we develop an algorithm to process queries depending on the types of nodes that the keywords match.
3. We conduct comprehensive experiments to demonstrate the effectiveness and efficiency of processing keyword queries using our ORM data graph approach over existing approaches.

2 Motivating Example

Let us consider the sample relational database in Figure 1. The relations *Student* and *Lecturer* store the core information about students and lecturers respectively. The qualifications of a lecturer are captured in the relation *Qualification* since each lecturer could have more than one qualification. The relation *Course* stores both the core information about courses and the many-to-one relationship between courses and lecturers. This reflects the application constraints that each course is associated with only one lecturer. The relation *Enrol* captures the many-to-many relationship between students and courses. The schema of this database can be modeled as a schema graph [10,9] where each node represents

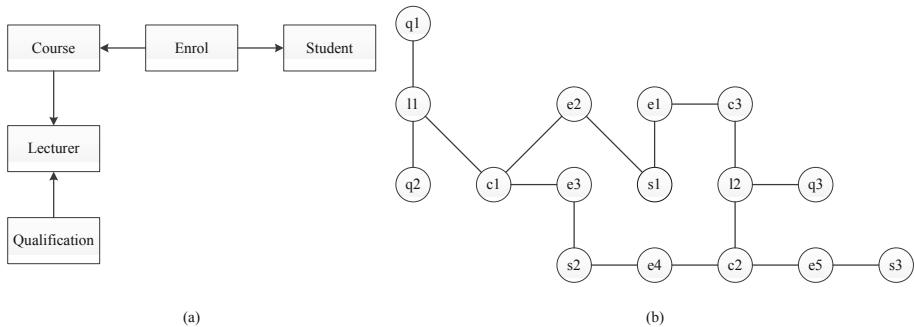


Fig. 2. The schema graph and the data graph for the example database in Fig. 1

a relation and each directed edge represents a foreign key-key constraint. Figure 2(a) shows the schema graph obtained. Correspondingly, the data instances of the database can be modeled as a data graph [11,8] where each node represents a tuple and each undirected edge represents a foreign key-key reference. As Figure 2(b) shows, the data graph is undirected as direction is not a major concern for query processing.

Example 1. Suppose a user issues the keyword query “*Steven Lee*” to retrieve all the information about him. Existing works will only return his *lid*, *name*, *office* and *email*, that is, the first tuple in the *Lecturer* relation. However, information about the degrees and associated majors and universities of “*Steven Lee*”, which are stored in the *Qualification* relation, is not retrieved.

Example 2. Suppose a user wants to know the information of the course where a student “*Mary*” obtains grade “*A*”, and issues the keyword query “*Mary A*”. Existing works will retrieve the third tuple in the *Student* relation and the last tuple in the *Enrol*, as the two query keywords occur in these tuples respectively and there exists a foreign key reference between them. This result is not informative as details such as the course id, title and credit is not retrieved.

In addition, relational keyword queries are also inherently ambiguous. Thus, existing works would consider all the possible interpretations of a keyword query and retrieve the corresponding information from the relational database. Consequently, a huge number of results are returned although many of them are probably not useful to the user.

Example 3. Suppose a user issues a keyword query “*John Mary*”. Figure 3 shows two sample results obtained by existing works. Intuitively, the first result shown in Figure 3(a) indicates that student “*John Williams*” is enrolled in the course “*Java Programming*” and student “*Mary Smith*” is enrolled in the course “*Information Retrieval*”. Both courses are taught by the same lecturer “*Janet Kate*”. The second result shown in Figure 3(b) means that student “*Edward Martin*” is enrolled in the same course “*Database Design*” as the student “*John Williams*”;

“Edward Martin” is also enrolled in the same course “Information Retrieval” as another student “Mary Smith”. We observe that the first result is most likely more useful to the user.

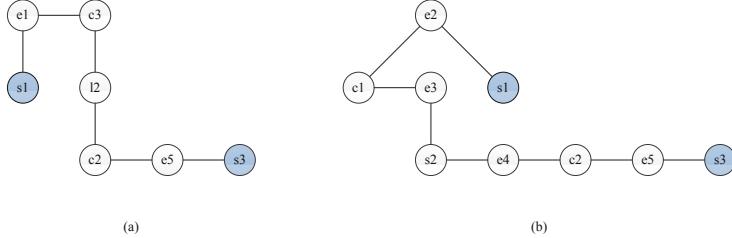


Fig. 3. Sample answers for query “*John Mary*” in Example 3

The above examples illustrate problems that arise when we do not consider the underlying semantics in the relational database. This motivates us to develop a semantic approach to answer keyword queries.

A relational database is typically designed using some conceptual model such as the ER diagram to capture the semantics in the real world in terms of entity and relationship types. Figure 4(a) shows the ER diagram for the relational database in Figure 1. The process of semantics discovery essentially reverses the translation from ER model to relational schema. There has been much research on discovering semantics from relational schema such as [19]. Here, we build upon these works and utilize primary key constraint and foreign key constraint to classify the relations in a relational schema.

Similar to [19], we have 4 types of relations, namely, *object relation*, *relationship relation*, *mixed relation* and *component relation*. Intuitively, an object (relationship) relation contains the majority of the attributes of an entity (relationship) type. A relation is a mixed relation if it encompasses both an entity type and a relationship type. A mixed relation occurs when there is a one-to-many relationship, e.g., the “Teach” relationship type in the ER diagram in Figure 4(a). A component relation represents a component part or the multi-valued attribute of an entity or relationship type, e.g, qualification is a multivalued attribute of Lecturer and is translated to the *Qualification* relation.

Based on the type of each relation in the database, we construct an *Object-Relationship-Mixed data graph* (ORM data graph) that consists of three types of nodes, namely object node (rectangle), relationship node (diamond) and mixed type node (hexagon). Each node typically includes some tuple in the corresponding relation. Tuples in the component relations are attached to their corresponding object (relationship or mixed) type nodes. In contrast to the traditional data graph where each node corresponds to a tuple in the database, a node in an ORM

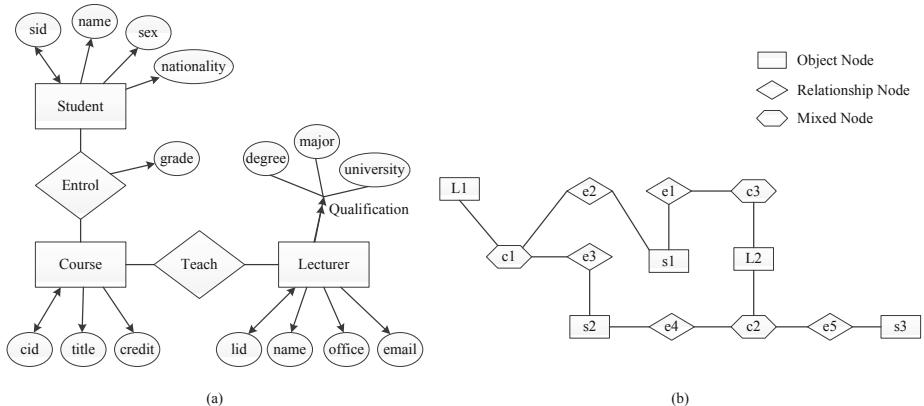


Fig. 4. The ER diagram and the ORM data graph for the database in Fig.1

data graph may correspond to a list of tuples. Two nodes are connected via an edge if there exists a foreign key-key reference between tuples in the nodes.

Figure 4(b) shows the ORM data graph¹ for the database in Figure 1. Node **L1** is an object node that includes the tuple **l1** in the object relation *Lecturer*. In addition, both tuple **q1** and **q2** in the component relation *Qualification* are associated with **l1** and attached to **L1**. Thus, node **L1** corresponds to a list of tuples $\{l1, q1, q2\}$. Node **c1** is a mixed type node that corresponds to the tuple **c1**. There is an edge between nodes **L1** and **c1** because of the foreign key-key reference between the tuples **l1** and **c1**.

We say that a query keyword matches a node in the ORM data graph if the keyword occurs in some tuple in the node. We devise two ways to process the query depending on the types of nodes that the keywords match. Consider the query “*Steven Lee*” in Example 1. Since both the keywords match object node **L1** in the graph, we will retrieve all the information about the lecturer object “*Steven Lee*”, including his qualifications. In other words, the three tuples $\{l1, q1, q2\}$ are returned as the answer.

For the keyword query “*John A*” in Example 2, the keyword “*A*” matches relationship node **e1**, while the keyword “*John*” matches object node **s1** that is directly connected with **e1**. Thus, we will return the information about the relationship along with all the participating objects, including the student object “*John*” and the course object “*Java Programming*”.

Finally, for Example 3, since the keywords “*John*” and “*Mary*” both match object nodes **s1** and **s3** in the ORM data graph, we will return a tree of nodes $\{s1 - e1 - c3 - L2 - c2 - e5 - s3\}$. Note that the node **L2** corresponds to a lecturer object which is common to both **s1** and **s3**.

¹ We use uppercase to label a node in the ORM data graph if it corresponds to a list of tuples, and lowercase if a node corresponds to a single tuple.

3 Proposed Approach

A keyword query Q is defined as $Q = \{k_1, k_2, \dots, k_n\}$, where $k_i, i \in [1, n]$ denotes a keyword. Each keyword is a term that specifies the user's search interest. Existing works consider that a keyword matches a tuple if the keyword is contained in the values of this tuple, and the goal of keyword query processing is to return the minimal number of tuples that collectively contain all the query keywords. While this approach retrieves all the tuples that contain the query keywords, the user will be overwhelmed with the large number of results. We observe that when a user issues a keyword query, it is usually directed at some object, or a relationship along with the associated objects.

Our proposed approach first utilizes key and foreign key constraints to classify the relations in a relational schema into *object*, *relationship*, *mixed* or *component* relations as follows:

1. A relation R is an *object relation* if there exists some relation R' that references R , and R does not reference other relations.
2. A relation R is a *relationship relation* if the primary key of R comprises more than one disjoint foreign key.
3. A relation R is a *mixed relation* if (a) there exists two relations R' and R'' such that R' references R and R references R'' , and (b) the primary key of R does not contain more than one disjoint foreign key.
4. A relation R_1 is a *component relation* if (a) no relation references R_1 , (b) the primary key of R_1 does not contain more than one disjoint foreign key, and (c) the inclusion dependency $R_1[A_1] \subseteq R[K]$ holds, where A_1 is a subset of attributes in R_1 and K is a candidate key of R .

A mixed relation contains information about both objects and relationships. We will use semantic dependencies to differentiate the objects and relationships when processing the keyword query. For example, the mixed relation $\text{Emp}(eno, ename, birthdate, address, dno, joindate)$ contains information about the employee and the date s/he joins a department. In this case, *joindate* is an attribute of the relationship between employee and department. This constraint can be captured by the *semantic dependency* [15] $\{eno, dno\} \xrightarrow{\text{Sem}} joindate$, indicating that the value of *joindate* will be updated when $\{eno, dno\}$ is updated. We consider the attributes *eno*, *ename*, *birthdate*, *address* the object part of the relation, and the attribute *joindate* the relationship part.

For each object relation R , we cluster the tuples in R and its component relations. Similarly, for each relationship (mixed) relation R , we also cluster the tuples in R and its component relations. Based on the clusters obtained, we construct an undirected *Object-Relationship-Mixed data graph (ORM data graph)* $G(V, E)$. Each node $v \in V$ corresponds to a cluster of tuples C . We have $v.label = C$, $v.tids$ is the list of tuple ids in cluster C , and $v.type \in \{\text{object}, \text{relationship}, \text{mixed}\}$ depending on whether tuples in the cluster are from an object relation, a relationship relation, or a mixed relation. An edge $e(u, v) \in E$ indicates a foreign key-key reference between tuples in u and v .

A query keyword k matches a node u in the ORM data graph G if k occurs in some tuple in u . Let $Obj(k)$ and $Rel(k)$ be the sets of object and relationship type nodes that match k respectively. Based on the semantic dependencies, if a keyword k matches the object part of a mixed type node u , then we add u to $Obj(k)$. Otherwise, if k matches the relationship part of u , we add u to $Rel(k)$.

If $Obj(k) \neq \emptyset$, that is, k matches some object type nodes and/or the object part of mixed type nodes, then we retrieve all the tuples associated with the nodes in $Obj(k)$. If $Rel(k) \neq \emptyset$, that is, k matches some relationship type nodes and/or the relationship part of mixed type nodes, then we retrieve the tuples associated with each node $v \in Rel(k)$, as well as the tuples in the object and mixed type nodes that are directly connected to v in the ORM data graph. The intuition is that when a keyword refers to some relationship, the user is either interested in the information about the relationship, or the information about the objects of the relationship. Thus, we will retrieve the information about the relationship, as well as the information about all the participating objects of this relationship.

After obtaining the tuples that match each keyword, we need to combine the results from different keyword matches. Given a keyword query Q , we have two main cases.

Case 1. $\exists k \in Q, Rel(k) \neq \emptyset$

For this case, the keywords in the query match either object, relationship or mixed type nodes. For each such k , we check each node $v \in Rel(k)$ whether the rest of the keywords match object and mixed type nodes that are directly connected to v in the ORM data graph. If so, then we return this result. We can view the result as a tree where the relationship node v is the root and the object and mixed type nodes are the leaves.

Recall the keyword query “*Mary A*” in Example 2. The keyword “*Mary*” matches object node $s3$, while keyword “*A*” matches relationship nodes $\{e1, e3, e5\}$ in the ORM data graph in Figure 4(b). Hence, we have $Obj(\text{“Mary”}) = \{s3\}$ and $Rel(\text{“A”}) = \{e1, e3, e5\}$. Since $s3$ and $e5$ are directly connected in the ORM data graph, we return the tuples associated with $e5$, as well as the tuples in $s3$ and $c2$ as the result. Intuitively, this result means that the student “*Mary Smith*” obtained grade “*A*” for the course “*Information Retrieval*”.

Case 2. $\forall k \in Q, Rel(k) = \emptyset$.

For this case, all the keywords match only object and mixed type nodes and we generate all the possible combinations of nodes from $Obj(k_1), Obj(k_2), \dots, Obj(k_{|Q|})$. For each node combination, we apply the standard graph traversal method to find the set of Steiner trees that connects these nodes. For each Steiner tree, we will check whether there exists a node v such that the path from each keyword matched node to v comprises of nodes from different relations in the schema. If so, we output this tree as a query result.

Recall our query “*John Mary*” in Example 3. Our algorithm will output the Steiner tree in Figure 3(a) but not in Figure 3(b) as the former contains node $l2$ such that both paths $l2 - c3 - e1 - s1$ and $l2 - c2 - e5 - s3$ comprises of nodes from different relations, while the latter does not contain a such node.

Algorithm 1 (ORMSearch) shows the details. The input is a keyword query Q , ORM data graph G and parameter K . We initialize two priority queues PQ_o and PQ_r to store candidate result trees ordered by the number of nodes in the tree (Line 1). For each keyword k , we find the set of nodes in G that match k . We partition the nodes into two sets: $Obj(k)$ and $Rel(k)$. For each node $v \in Rel(k)$, we create a tree $T_{v,k}$ that consists of v and its neighboring nodes in the ORM data graph G . $T_{v,k}$ is associated with the keyword k to denote that k matches some node in the tree. If the tree already exists in queue PQ_r , we update the associated keywords of the tree by adding k . Otherwise, we insert the tree into queue PQ_r (Lines 4-6). Similarly, for each node $v \in Obj(k)$, we create a tree $T_{v,k}$ that consists of a root node v . If the tree exists in the queue PQ_o , we update the associated keywords of the tree by adding k . Otherwise, we insert the tree into PQ_o (Lines 8-10).

Next, we combine the results from different keyword matches. Lines 11-24 process the trees in PQ_r (Case 1). We initialize a variable $count$, and iteratively dequeue a tree T from PQ_r . We obtain the set of keywords W associated with T (Lines 13-14). For each query keyword k that does not appear in W , we check whether k matches some node in T . If so, we put k into W (Lines 15-17). Finally, if every query keyword matches some node in T , we will put T into $Result$ and increase $count$ (Lines 19-20). This process terminates when $count$ equals to K , i.e., we have already found K number of results (Lines 21-22).

Lines 25-48 process the trees in PQ_o (Case 2). For each iteration, we dequeue a tree T from PQ_o . Let v be the root of T and W be the set of keywords associated with T . If every keyword matches some node in T , we put T into $Result$ and increase $count$ (Lines 27-30). If $count$ equals to K , we exit the loop. Otherwise, we traverse the ORM data graph G to find the set of Steiner trees that associate all the query keywords. We use *tree grow* and *tree merge* strategies in [6] to expand Steiner trees associated with partial keywords to those associated with all query keywords.

For each node u that is directly connected to v in G , we create a new tree T' from T by adding u as the new root of T' (Lines 36-37). This process is called tree growing. We first check whether there exists a node y in the new tree T' such that every path from y to a leaf node consists of nodes from distinct relations. If so, then we check if PQ_o already contains a tree with root u and associated with keywords W . If yes, then we update PQ_o with the smaller tree, else we insert T' into PQ_o (Lines 38-40).

For each set of keywords W' such that W' is a subset of Q and W' has no common keywords with W , we check whether we have found a tree T' with root v and associated with keywords W' in previous iterations. If T' exists, we create a new tree T'' by merging T' and T . T'' is rooted at v and associated with keywords $W \cup W'$ (Lines 42-44). This process is called tree merging. After that, we check whether there exists a node y in the new tree T'' such that every path from y to a leaf node consists of nodes from distinct relations. If so, we update queue PQ_o with T'' (Lines 45-47). Finally, we return the top- K trees in $Result$ (Line 49).

Algorithm 1. ORMSearch

```

input : keyword query  $Q = \{k_1, \dots, k_n\}$ ,  $K$ , ORM data graph  $G$ ,
output: result set  $Result$ 
1  $Result \leftarrow \emptyset; PQ_o \leftarrow \emptyset; PQ_r \leftarrow \emptyset;$ 
2 for  $i = 1$  to  $n$  do
3   Let  $Rel(k_i)$  be the set of relationship/mixed nodes in  $G$  that match  $k_i$ ;
4   foreach node  $v \in Rel(k_i)$  do
5     create a tree  $T_{v,k_i}$  that consists of  $v$  and its neighboring nodes in  $G$ ;
6     update  $PQ_r$  with  $T_{v,k_i}$ ;
7   Let  $Obj(k_i)$  be the set of object/mixed nodes in  $G$  that match  $k_i$ ;
8   foreach node  $v \in Obj(k_i)$  do
9     create a tree  $T_{v,k_i}$  with root  $v$ ;
10    update  $PQ_o$  with  $T_{v,k_i}$ ;

11 count = 0;
12 while  $PQ_r \neq \emptyset$  do
13    $T = \text{dequeue } PQ_r;$ 
14   Let  $W$  be the set of keywords that are associated with  $T$ ;
15   foreach keyword  $k \in Q - W$  do
16     if  $k$  matches some node in  $T$  then
17       |  $W = W \cup \{k\}$ ;
18
19   if  $W == Q$  then
20     add  $T$  to  $Result$ ; count++;
21     if  $count = K$  then
22       | break;
23
24
25 count = 0;
26 while  $PQ_o \neq \emptyset$  do
27    $T = \text{dequeue } PQ_o;$ 
28   let  $v$  be the root of  $T$  and  $W$  be the set of keywords associated with  $T$ ;
29   if  $W == Q$  then
30     add  $T$  to  $Result$ ; count++;
31     if  $count = K$  then
32       | break;
33
34 else
35   //Tree growing process
36   foreach node  $u$  that is directly connected to  $v$  in  $G$  do
37     create a new tree  $T'$  from  $T$  by adding  $u$  as the new root;
38     if  $\exists$  node  $y \in T'$  s.t. every path from  $y$  to a leaf node consists of nodes from
39     distinct relations then
40       | update  $PQ_o$  with  $T'$ ;
41
42   //Tree merging process
43   foreach set of keywords  $W' \subset Q$  s.t.  $W \cap W' = \emptyset$  do
44     if  $\exists$  tree  $T'$  s.t.  $v$  is the root of  $T'$  and  $W'$  is the set of keywords associated
45     with  $T'$  then
46       merge  $T'$  with  $T$  to form  $T''$ ;
47       if  $\exists$  node  $y \in T''$  s.t. every path from  $y$  to a leaf node consists of nodes
48       from distinct relations then
49         | update  $PQ_o$  with  $T''$ ;

49 return the top- $K$  trees in  $Result$ ;

```

4 Performance Study

In this section, we evaluate the effectiveness and the efficiency of our semantic approach. We adopt the traditional data graph approach as the baseline because our approach also performs search directly on the data. We use the well-established Steiner tree and the state of the art DPBF [6] implementation. Since the ranking of results is orthogonal to this work, we will output query results ordered by the number of nodes in the result.

Two real world datasets are used in our experiments: the Internet Movie Database (IMDB)² and the DBLP data (DBLP) [5]. For the IMDB dataset, we convert a subset of its raw text files into 8 relations. The total number of tuples is 2,168,813. For the DBLP dataset, the schema consists of 6 relations and the data consists of 881,867 tuples. Table 1 shows the keyword queries used.

The experiments were performed on a Intel(R) Core(TM) i7-2600 CPU 3.40GHz with 8GB of RAM. All the algorithms were implemented using JDK 1.7 and JDBC. The inverted indices are built using MySQL v5.5 fulltext index.

Table 1. Queries used in experiments

DBLP	IMDB
DQ1 Keyword Search	IQ1 Christopher Nolan
DQ2 SIGMOD Jeffrey	IQ2 Woody Allen
DQ3 Jim Gray Alexander	IQ3 Johnny Depp Jack
DQ4 PageRank Computing research	IQ4 Jamie Paul Jones
DQ5 Query optimization Yannis Papakonstantinou	IQ5 Steven Horse drama
DQ6 Conceptual design relational database	IQ6 Peter Parker comedy
DQ7 Ling Tok Wang Object Relationship	IQ7 American Comedy Page Ellen

4.1 Effectiveness Experiments

We first compare the query results returned by ORMSearch and DPBF. Table 2 shows a sample of the results for the IMDB dataset. We observe that the results obtained by DPBF is not as informative as those obtained by ORMSearch. *Q1* is a query about the movie “Inception”. ORMSearch retrieves all the information about this movie but DPBF does not retrieve the genre information. *Q2* is a query about the movie “Intouchables” and the character name “Nouvel”. Compared to ORMSearch, DPBF provides no information about the actor who played the character “Nouvel” in “Intouchables”. For *Q3*, ORMSearch retrieves movies where Jeremy plays the character Cruise, as well as movies where both Jeremy and Cruise act in. In contrast, DPBF retrieves 174 results, many of which are not useful. The results for DBLP is similar and we omit it due to space limit.

Figure 5(a) and 5(b) show the number of results retrieved for each query on both datasets when we set the result size to 7 and 9 respectively. We see that DPBF typically produces more results than ORMSearch. Moreover, when the result size increases from 7 to 9, the number of results returned by DPBF increases significantly.

² <http://www.imdb.com/interfaces>

Table 2. Results of queries for IMDB dataset

Query	ORMSearch	DPBF
Q1: Inception	1. Movie: Inception 2010 Action Adventure Mystery	1. Movie: Inception 2010
Q2: Intouchables Nouvel	1. Movie: Intouchables 2011 Comedy Drama Character: Nouvel auxiliaire 2 Actor: Cayrey, Jean Fran	1. Movie: Intouchables 2011 Character: Nouvel auxiliaire
Q3: Cruise Jeremy	1. Movie: Car Jack 2008 Action Adventure Crime Character: Cruise Actor: Anus, Jeremy 2. Movie: August 2008 Drama Actor: Bobb, Jeremy Actor: Cruise, Tom 3. Movie: Mission: Impossible-Ghost Protocol 2011 Actor: Renner, Jeremy Actor: Cruise, Tom	1. Character: Cruise Actor: Anus, Jeremy 2. Character: Cruise Guy Actor: Palko, Jeremy 3. Movie: Knocked Down 2008 Character: Irving Cruise Character: Cab Driver Actor: Aimone, Jeremy ...

To further verify that our approach can achieve a better search quality than the base line, we carried out a survey where we show the queries to 6 users and collect the possible search intentions (at most 5) of each query. For each particular search intention, we generate an SQL statement and take the SQL execution results. Results of all the SQLs form the ground truth for us to determine the precision of the results obtained by ORMSearch and DPBF. Figure 5(c) and 5(d) show that ORMSearch is able to achieve a much higher precision than DPBF for most of the queries. Both ORMSearch and DPBF has a precision of 1.0 for query *DQ2* as it has only one possible search intention. The precision of DPBF is low for query *DQ6* as it is inherently ambiguous with a large number of possible search intentions. However, ORMSearch is still able to improve the precision by retrieving more informative and useful results.

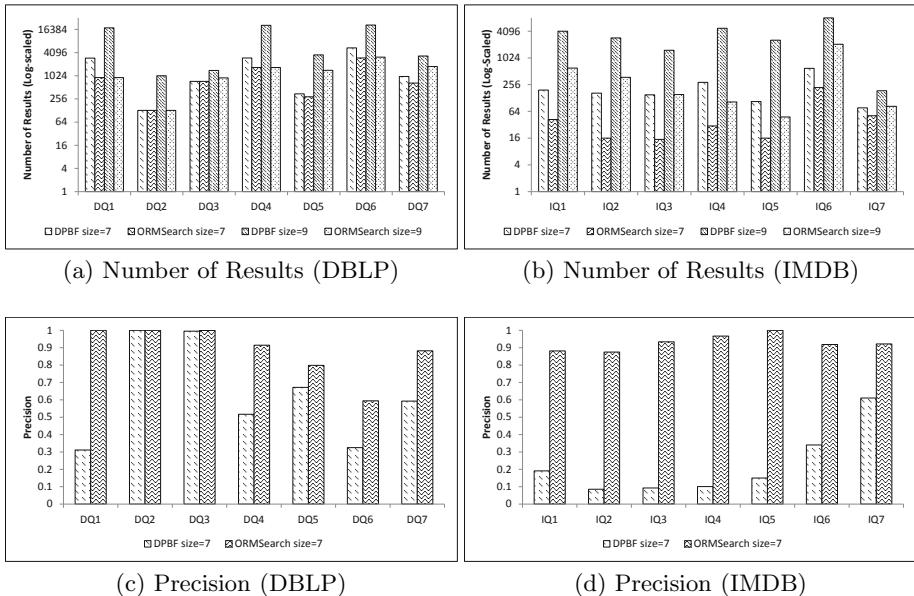
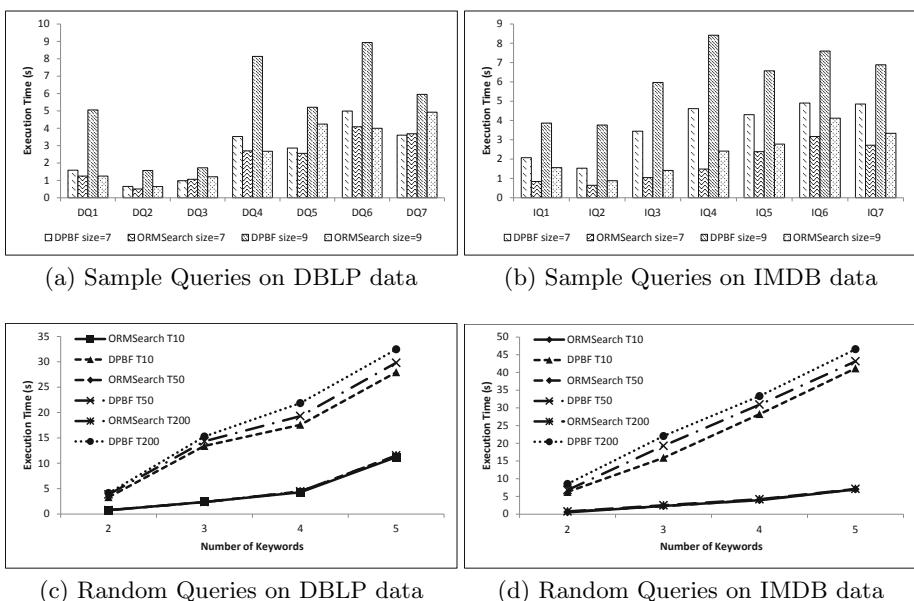
4.2 Efficiency Experiments

Finally, we compare the execution time of the two approaches. Figure 6(a) and Figure 6(b) show the results. As we can see, ORMSearch is about 2~3 times faster than DPBF, especially when the maximum result size is 9.

Besides the queries in Table 1, we also randomly generate 40 queries for each dataset whose lengths vary from 2 to 5 keywords, with 10 queries for each query size. For each query, we test the execution time of ORMSearch and DPBF for retrieving first output 10, 50 and 200 results respectively. The average execution time on cold cache is recorded in Figure 6(c) and Figure 6(d). On average, ORMSearch is about 6~8 times faster than DPBF. Further, the time required by ORMSearch to retrieve 10, 50, and 200 results are almost the same, while the execution time for DPBF increases. The gap between ORMSearch and DPBF widens as the number of keywords increases. This is because our ORM data graph has fewer nodes compared to the traditional data graph.

5 Related Work

Relational keyword search can be broadly classified into two categories: (a) schema graph approach and (b) data graph approach. In the schema graph

**Fig. 5.** Effectiveness of ORMSearch vs DPBF**Fig. 6.** Efficiency of ORMSearch vs DPBF

approach, the database schema is modeled as an undirected graph where each node represents a relation and each edge represents a foreign key-key constraint. To answer a keyword query, DBXploer [1] proposes join trees such that each leaf relation covers some keyword with tuples containing that keyword, and all the leaf relations collectively cover all keywords of the query. Thus, by joining all the relations in a join tree, the output tuples will contain all keywords specified in the query. Discover [10] tries to find join trees without any redundant leaf relations that cover the same keywords as others. In addition, it allows duplicate relations in join trees because a relation can join itself via a many-to-many relationship with other relations. [9] is a variant of Discover, which relaxes the requirement that the output tuples should contain all the keywords in a query.

In the data graph approach, the relational database is modeled as a graph where each node corresponds to a tuple and each edge corresponds to a foreign key-key reference. Banks [11] proposes a backward expansion search to find the common node which connects a keyword node for each keyword via the shortest path. An answer to a query is an Steiner tree with the common node as the root and each keyword node as a leaf. [12] improves the performance of [11] by using a bidirectional expansion technique to reduce the size of the search space. [6] proposes dynamic programming to identify the top-k minimal group Steiner tree in time exponential in the number of keywords. [8] proposes a bidirectional index to improve query performance. [14] studies how to calculate the radius of a graph and defines an answer to a keyword query as a subgraph which has a user-specified radius and is relevant to each keyword. These works are focused on the efficiency of relational keyword search and do not consider the quality of the search results.

To improve the search quality, [9] adopts an IR-style ranking strategy to evaluate the relevance of an answer. [16] further improves the search effectiveness by normalizing the ranking formulae in [9]. Spark [17] considers all the tuples in the answer as a visual document to avoid the side effect of overly rewarding contributions of the same keyword. Banks [11] evaluates the relevance of an answer tree by investigating its root and each leaf nodes. [20] measures the importance of not only the root and leaf nodes, but also the intermediate nodes. However, none of these works addresses the problems raised in Section 2.

Objectrank [2] considers the database as a set of objects. However, it is not clear how database objects are detected. [18] proposes to infer the basic, independent semantic unit of information in a database, but does not consider the relationship between semantic units. [7] defines a query result as an object summary (OS) about a particular data subject. The relationship between data subjects is not considered. [3] proposed a statistical way to find the promising search target(s) for an XML keyword query.

6 Conclusion

In this paper, we have examined the limitations of existing relational keyword search methods, and proposed a semantic approach to address the problems of

retrieving informative and useful results. This is achieved by constructing an ORM data graph to capture the semantics of objects and relationships in the database. Compared to the traditional data graph, each node in the ORM data graph is associated with a type and may correspond to a list of tuples. Based on the ORM data graph, we devised an efficient algorithm to process keyword queries. Experiments on two real world datasets verify the effectiveness and efficiency of our approach.

References

1. Agrawal, S., Chaudhuri, S., Das, G.: DBXplorer: A system for keyword-based search over relational databases. In: ICDE (2002)
2. Balmin, A., Hristidis, V., Papakonstantinou, Y.: Objectrank: authority-based keyword search in databases. In: VLDB (2004)
3. Bao, Z., Ling, T.W., Chen, B., Lu, J.: Effective xml keyword search with relevance oriented ranking. In: ICDE (2009)
4. Bergamaschi, S., Domnori, E., Guerra, F., Trillo Lado, R., Velegrakis, Y.: Keyword search over relational databases: a metadata approach. In: SIGMOD (2011)
5. Cyganiak, R.: D2RQ benchmarking,
<http://sites.wiwiiss.fu-berlin.de/suhl/bizer/d2rq/benchmarks/>
6. Ding, B., Yu, J.X., Wang, S., Qin, L., Zhang, X., Lin, X.: Finding top-k min-cost connected trees in databases. In: ICDE (2007)
7. Fakas, G.J., Cai, Z., Mamoulis, N.: Size-l object summaries for relational keyword search. Proc. VLDB Endow. (2011)
8. He, H., Wang, H., Yang, J., Yu, P.S.: BLINKS: ranked keyword searches on graphs. In: SIGMOD (2007)
9. Hristidis, V., Gravano, L., Papakonstantinou, Y.: Efficient IR-style keyword search over relational databases. In: VLDB (2003)
10. Hristidis, V., Papakonstantinou, Y.: Discover: keyword search in relational databases. In: VLDB (2002)
11. Hulgeri, A., Nakhe, C.: Keyword searching and browsing in databases using banks. In: ICDE (2002)
12. Kacholia, V., Pandit, S., Chakrabarti, S.: Bidirectional expansion for keyword search on graph databases. In: VLDB (2005)
13. Kargar, M., An, A.: Keyword search in graphs: finding r-cliques. Proc. VLDB Endow. (2011)
14. Li, G., Ooi, B.C., Feng, J.: EASE: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In: SIGMOD (2008)
15. Ling, T.W., Lee, M.L.: Relational to entity-relationship schema translation using semantic and inclusion dependencies. Integr. Comput.-Aided Eng. (1995)
16. Liu, F., Yu, C., Meng, W., Chowdhury, A.: Effective keyword search in relational databases. In: SIGMOD (2006)
17. Luo, Y., Lin, X., Wang, W., Zhou, X.: Spark: top-k keyword query in relational databases. In: SIGMOD (2007)
18. Nandi, A., Jagadish, H.V.: Qunits: queried units for database search. In: CIDR (2009)
19. Yan, L.-L., Ling, T.W.: Translating relational schema with constraints into OODB schema. In: Database Semantics Conference (1993)
20. Yu, X., Shi, H.: CI-Rank: Ranking keyword search results based on collective importance. In: ICDE (2012)

Semantic-Based Mappings

Giansalvatore Mecca¹, Guillem Rull²,
Donatello Santoro^{1,3}, and Ernest Teniente²

¹ Università della Basilicata – Potenza, Italy

² Universitat Politècnica de Catalunya – Barcelona, Spain*

³ Università Roma Tre – Roma, Italy

Abstract. Data translation consists of the task of moving data from a source database to a target database. This task is usually performed by developing mappings, i.e., executable transformations from the source to the target schema. However, it is often the case that a richer description of the target database semantics is available under the form of a conceptual schema. We investigate how the mapping process changes when such a rich conceptualization of the target database is available. As a major contribution, we develop a translation algorithm that automatically rewrites a mapping from the source database schema to the target conceptual schema into an equivalent mapping from the source schema to the underlying target database schema. Experiments show that our approach scales nicely to complex conceptual schemas and large databases.

1 Introduction

Integrating data coming from disparate sources is a crucial task in many applications. An essential requirement of any data integration task is that of manipulating *mappings* between sources. Mappings are executable transformations which define how an instance of a source repository should be translated into an instance of a target repository. Traditionally, mappings are developed to exchange data between two relational database schemas [15]. A rich body of research has been devoted to developing algorithms to simplify the specification of the mapping [19], formalizing the semantics of the translation process [8], and improving the quality of results [23,13,12].

This paper investigates how the mapping process changes in presence of richer *conceptual schemas* of the two data sources. In fact, the repositories used in the organization by the various processes and applications often undergo modifications during the years, and may lose their original design. As a consequence, it is often the case that an additional description of the domain of interest is available under the form of a (*mediator*) conceptual schema. The global unified view given by the mediator schema is constructed independently from the representation adopted for the data stored at the sources. Also, these semantic conceptual schemas are particularly important in the context of information integration since

* This work has been partially supported by the Ministerio de Ciencia y Tecnología under project TIN2011-24747.

they are used to provide a transparent access through a global schema to a collection of data stored in multiple heterogeneous data sources [11]. It is therefore important to study how the mapping process changes in this setting.

We assume that a mediator conceptual schema is provided for the target and, possibly, for the source data repository. The relationship between the domain concepts in this conceptual schema and the data sources is given by a set of views, which define the conceptual constructs in terms of the logical database tables using a relational language of conjunctive queries, comparisons and negations, as discussed in the following example.

Motivating Example. Assume we have the two relational schemas below and we need to develop a mapping to translate data among them (from the source to the target).

Source schema:	<i>Emp(id, name, dept, rating)</i>	<i>Dept(name, manager)</i>
Target schema:	<i>Employee(id, name, dept)</i>	<i>Dept(name)</i>
	<i>HasPassedTest(emp, testcode)</i>	<i>DisciplinaryProcedure(pnum, emp)</i>
	<i>Manager(mgr, dept)</i>	

Clearly, both schemas rely on the same domain, which includes data from employees and the departments they work in. However, it is not evident at all how the source-to-target mapping should be defined since it is difficult to establish a clear correspondence between the tables in the source schema and those in the target. This is so mainly because the target schema contains some tables, such as *HasPassedTest* or *DisciplinaryProcedure*, whose contents is difficult to relate to the information stored in the tables *Emp* and *Dept* from the source schema.

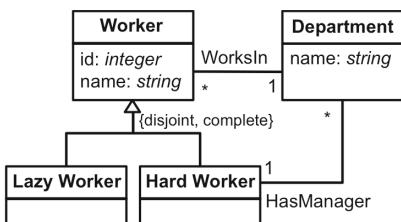


Fig. 1. Target conceptual schema

of this mediator conceptual schema is closer to the way the information is stored in the source schema than the one provided by the physical target tables. Therefore, the mapping designer will find it easier to define the mapping from the

```

Worker(id, name) ← Employee(id, name, dept)
Promoted(id) ← HasPassedTest(id, tc), ¬DisciplinaryProcedure(pnum, id)
LazyWorker(id, name) ← Employee(id, name, dept), ¬Promoted(id)
HardWorker(id, name) ← Employee(id, name, dept), ¬LazyWorker(id, name)
Department(name) ← Dept(name)
WorksIn(id, dept) ← Employee(id, name, dept)
HasManager(dept, mgr) ← Manager(mgr, dept)
  
```

Fig. 2. View definitions for the target conceptual schema

Suppose now that the mediator conceptual schema corresponding to the target relational schema is the one shown in Figure 1. The views defining each class and association in the conceptual schema in terms of the database tables are reported in Figure 2 (keys are underlined). Notice how, to improve the expressibility of the language, negated literals are used in the body of view definitions, that is, the view-definition language is non-recursive Datalog [6] with negation. The semantics

source schema to the target conceptual schema. For instance, he could easily realize that lazy workers in the target conceptual schema are those employees in the source with a rating lower than 5, while the rest of employees are hard workers. As it is common [8], we shall use *tuple generating dependencies* (*tgds*) and *equality-generating dependencies* (*egds*) [3] to express the mappings. In our case, the translation of source tuples into the **LazyWorkers** and **HardWorkers** target concepts can be expressed by using the following *tgds* with comparison atoms:

$$\begin{aligned} m_0 : \forall id, name, dept, rat : Emp(id, name, dept, rat), rat < 5 \rightarrow \text{LazyWorker}(id, name) \\ m_1 : \forall id, name, dept, rat : Emp(id, name, dept, rat), rat \geq 5 \rightarrow \text{HardWorker}(id, name) \end{aligned}$$

Intuitively, mapping expression m_0 specifies that, for each tuple in the source table *Emp* such that the *rating* attribute is lower than 5, there should be a lazy worker in the target. Similarly for m_1 . Mappings m_2 below relates the *Dept* source table to the class **Department** and the association **HasManager**, and m_3 relates table *Emp* to association **WorksIn** (from now on, we omit universal quantifiers):

$$\begin{aligned} m_2 : Dept(name, mgr) \rightarrow \text{Department}(name), \text{HasManager}(name, mgr) \\ m_3 : Emp(id, name, dept, rating) \rightarrow \text{WorksIn}(id, dept) \end{aligned}$$

Suppose now we want to perform the exchange and actually bring data from the source database to the target. Unfortunately, mappings $m_0 - m_3$ above are not directly executable. In fact, they refer to virtual entities – the constructs in the conceptual schema – and not to the actual tables in the target. We therefore need to devise a way to translate such source-to-semantic mapping into a classical source-to-target one (which copies data directly from the source into the target database) in order to be able to execute it. That is, given a source-to-semantic mapping, the target conceptual schema, and the views defining this schema in terms of the underlying database, we want to obtain the corresponding source-to-target mapping.

Contributions. This paper develops a number of techniques to solve this kind of *semantic-based mapping problems*. More specifically:

- (i) we develop rewriting algorithms to automatically translate mappings over the semantic schema into mappings over the underlying databases; we first discuss the case in which a semantic schema is available for the target database only; then, we extend the algorithm to the case in which a semantic schema is available both for the source and the target;
- (ii) the algorithm that rewrites a source-to-semantic mapping into a classical and executable source-to-target mapping is based on the idea of unfolding views in mapping conclusions; however, in our setting, this unfolding is far from being straightforward; in the paper, we show that the problem is made significantly more complex by the expressibility of the view definition language, and more precisely, by the presence of negated atoms in the body of view definitions; we investigate the implications of adopting such an expressive language, and propose a solution that represents a good compromise between expressibility and complexity;
- (iii) the classical approach to executing a source-to-target exchange consists in running the given mappings using a *chase* engine [8]; we develop a chase engine

for this task, and integrate it within the working prototype of our semantic-based mapping system; using the prototype, we conduct several experiments to show that our approach scales nicely to large databases and mapping scenarios.

We believe this paper represents a significant step forward towards the goal of incorporating richer descriptions, under the form of conceptual schemas, into the data translation process. Given the evolution of the Semantic Web, and the increased adoption of ontologies, we believe this represents an important problem that may open up further research directions.

The paper is organized as follows. Section 2 recalls some basic notions and definitions. Section 3 introduces the semantic-based mapping problem. The rewriting algorithm and formal results are in Section 4. Experiments are in Section 5. Related works are in Section 6.

2 Background

Conceptual Schemas. A *conceptual schema* in information systems is the general knowledge that the system needs about its domain in order to perform its functions [17]. In this paper, we focus on the part of a conceptual schema that deals with static aspects, i.e., the *structural schema*. In particular, we consider structural schemas that consist of a taxonomy of entity types (which may have attributes), a taxonomy of relationship types (defined among entity types), and a set of integrity constraints (which affect the state of the domain). The integrity constraints are expressed by means of *dependencies* (see subsection below). Throughout the paper, whenever we use the term conceptual schema, we are referring to a structural schema that conforms to this language.

Databases. We focus on the relational setting. A *schema* \mathbf{S} is a set of relation symbols $\{R_1, \dots, R_n\}$ each with an associated relation schema $R(A_1, \dots, A_m)$. Given schemas \mathbf{S}, \mathbf{T} with disjoint relations symbols, $\langle \mathbf{S}, \mathbf{T} \rangle$ denotes the schema corresponding to the union of \mathbf{S} and \mathbf{T} . An *instance* of a schema is a set of tuples in the form $R(v_1, \dots, v_m)$, where each v_i denotes either a constant, typically denoted by a, b, c, \dots , or a *labeled null*, denoted by N_1, N_2, \dots . Constants and labeled nulls form two disjoint sets. Given instances I and J , a homomorphism $h : I \rightarrow J$ is a mapping from $\text{dom}(I)$ to $\text{dom}(J)$ such that for every $c \in \text{CONST}$, $h(c) = c$, and for all tuple $t = R(v_1, \dots, v_n)$ in I , it is the case that $h(t) = R(h(v_1), \dots, h(v_n))$ belongs to J . Homomorphisms immediately extend to formulas, since atoms in formulas can be seen as tuples whose values correspond to variables.

Views. To bridge the gap between the conceptual schema and the underlying database, we assume a set of GAV views (Global-As-View) is given, i.e., we assume there is one view for each entity and relationship type which defines this type in terms of the underlying database (see Figure 2). A *view* V is a derived relation defined over a schema S . The view definition for V over S is a non-recursive rule in the form of $V(\bar{x}) \Leftarrow R_1(\bar{x}_1), \dots, R_p(\bar{x}_p), \neg R_{p+1}(\bar{x}_{p+1}), \dots, \neg R_{p+g}(\bar{x}_{p+g})$, with $p \geq 1$ and $g \geq 0$, where the variables in \bar{x} are taken from $\bar{x}_1, \dots, \bar{x}_p$.

Atoms in a view definition can be either base or derived. An atom $V(\bar{x})$ is a *derived atom* if V denotes a view; otherwise, it is a *base atom*. A view definition specifies how the extension of the view is computed from a given instance of the underlying schema, that is, given a homomorphism h from the definition of V to an instance I , $h(V(\bar{x}))$ belongs to the extension of V iff $h(R_1(\bar{x})) \wedge \dots \wedge \neg h(R_{p+g}(\bar{x}_{p+g}))$ is true on I .

Dependencies. A *tuple-generating dependency (tgd)* over \mathbf{S} is a formula of the form $\forall \bar{x}, \bar{z} (\varphi(\bar{x}, \bar{z}) \rightarrow \exists \bar{y} \psi(\bar{x}, \bar{y}))$, where $\varphi(\bar{x}, \bar{z})$ and $\psi(\bar{x}, \bar{y})$ are conjunctions of atoms. We allow two kinds of atoms in the premise: (a) relational atoms over \mathbf{S} ; (b) comparison atoms of the form $v \text{ op } c$, where op is a comparison operator ($=, >, <, \geq, \leq$), v is a variable that also appears as part of a relational atom, and c is a constant. Only relational atoms are allowed in the conclusion. A *denial constraint* is a special form of tgd of the form $\forall \bar{x} (\varphi(\bar{x}) \rightarrow \perp)$, in which the conclusion only contains the \perp atom, which cannot be made true. An *equality generating dependency (egd)* over \mathbf{S} is a formula of the form $\forall \bar{x} (\phi(\bar{x}) \rightarrow x_i = x_j)$ where $\phi(\bar{x})$ is a conjunctions of relational atoms over \mathbf{S} and comparison atoms as defined above, and x_i and x_j occur in \bar{x} .

Mapping Scenarios. A *mapping scenario*, $\mathcal{M} = \{\mathbf{S}, \mathbf{T}, \Sigma_{ST}, \Sigma_T\}$, is a quadruple consisting of a source schema \mathbf{S} , a target schema \mathbf{T} , and a set of *source-to-target tgds* Σ_{ST} – i.e., tgds such that the premise is a formula over \mathbf{S} and the conclusion a formula over \mathbf{T} –, and Σ_T is a set of *target tgds* – tgds over \mathbf{T} – and *target egds* – egds over \mathbf{T} . Given a source instance I , a solution for I under \mathcal{M} is a target instance J such that I and J satisfy Σ_{ST} , and J satisfies Σ_T . A solution J for I and \mathcal{M} is called a *universal solution* if, for all other solution J' for I and \mathcal{M} , there is a homomorphism from J to J' . The chase is a well-known algorithm for computing universal solutions [8].

3 The Semantic-Based Mapping Problem

In this section we shall introduce our mapping problem. As we discussed above, let us first assume that a conceptual schema is only available for the target database. Later on we shall discuss how things can be extended to handle a source conceptual schema as well.

Source-to-Semantic Mappings. The inputs to our source-to-semantics mapping problem are: (i) a source relational schema, \mathbf{S} , and a target relational schema \mathbf{T} ; (ii) a target conceptual schema, \mathbf{T}' , defined by means of a set of view definitions, Σ_V , over \mathbf{T} , as in Figure 2. View definitions may involve negations over derived atoms; (iii) a set of egds, $\Sigma_{T'}$, to encode key constraints and functional dependencies over the conceptual schema; (iv) finally, as an additional input of crucial importance, we assume the definition of the source-to-semantic mapping, $\Sigma_{ST'}$ as a set of s-t tgds over \mathbf{S} and \mathbf{T}' .

In the following, we always assume that the input mapping captures all of the semantics from the conceptual level. To do this, we assume the graphical integrity constraints of the conceptual schema are properly encoded into the mapping [9].

We restrict the textual integrity constraints to be key constraints and functional dependencies, and assume they are expressed as logic dependencies over the views (an automatic OCL-to-logic translation is proposed in [20]). Therefore, the complete set of dependencies $\Sigma_{ST'}$ and $\Sigma_{T'}$ for our running example is reported in Figure 3.

$$\begin{aligned}
 m_0 : & Emp(id, name, dept, rating), rating < 5 \rightarrow \text{LazyWorker}(id, name), \text{Worker}(id, name), \\
 & \text{WorksIn}(id, dept), \text{Department}(dept) \\
 m_1 : & Emp(id, name, dept, rating), rating \geq 5 \rightarrow \text{HardWorker}(id, name), \text{Worker}(id, name) \\
 & \text{WorksIn}(id, dept), \text{Department}(dept) \\
 m_2 : & Dept(name, mgr) \rightarrow \exists mname : \text{Department}(name), \text{HasManager}(name, mgr), \\
 & \text{HardWorker}(mgr, mname), \text{Worker}(mgr, mname) \\
 m_3 : & \text{Worker}(id, name), \text{Worker}(id, name') \rightarrow name = name' \\
 m_4 : & \text{WorksIn}(id, dept), \text{WorksIn}(id, dept') \rightarrow dept = dept' \\
 m_5 : & \text{HasManager}(dept, mgr), \text{HasManager}(dept, mgr') \rightarrow mgr = mgr'
 \end{aligned}$$

Fig. 3. Source-to-semantic mapping

Based on these, our intention is to rewrite the dependencies in $\Sigma_{ST'} \cup \Sigma_{T'}$ as a new set of dependencies, from the source to the target database. More specifically, we shall generate:

- (a) a new set of source-to-target tgds, Σ_{ST} ;
- (b) a set of target dependencies, Σ_T . This latter set will contain: (b.1) a set of target egds, as it is expected, in order to model egds over the conceptual schema. However, it may also incorporate other constraints that were not in the input. More precisely: (b.2) target tgds, i.e., tgds defined over the symbols in the target only; (b.3) denial constraints. Recall from Section 2 that a denial constraint is a dependency of the form $\forall \bar{x}(\varphi(\bar{x}) \rightarrow \perp)$. In our approach, these are used to express the fact that some tuple configurations in the target are not compatible with the view definitions, and therefore should cause a failure in the mapping process.

The process is illustrated in Figure 5.a, where solid lines refer to inputs, and dashed lines to outputs produced by the rewriting. We shall require that the result of executing the source-to-target mapping is “equivalent” to the one that we would obtain if the source-to-semantic mapping was to be executed. By equivalent we mean that a universal solution produced by the source-to-target mapping induces a universal solution for the source-to-semantic mapping when applying the view definitions. To be more precise, we first consider the source-to-semantic mapping scenario: $\mathcal{M}_{ST'} = \{\mathbf{S}, \mathbf{T}', \Sigma_{ST'}, \Sigma_{T'}\}$. For each source instance I , assume there exist a universal solution $J_{T'}$ for I and $\mathcal{M}_{ST'}$ that complies with the view definitions in Σ_V (i.e., there exist an instance J_T of schema \mathbf{T} such that $J_{T'} = \Sigma_V(J_T)$). Figure 4.a and b show one example of I and $J_{T'}$.

We compute our rewriting, and obtain a new source-to-target scenario: $\mathcal{M}_{ST} = \{\mathbf{S}, \mathbf{T}, \Sigma_{ST}, \Sigma_T\}$. We may run \mathcal{M}_{ST} on I to obtain solutions under the form of target instances. To any target instance J_T of this kind, we may apply the view definitions in Σ_V in order to obtain an instance of \mathbf{T}' , $\Sigma_V(J_T)$.

a. Source instance I		
$Emp(1, John, Marketing, 5)$	$Emp(2, Luke, Shipping, 7)$	$Emp(3, Mike, Shipping, 2)$
$Dept(Marketing, 1)$	$Dept(Shipping, 2)$	
b. Conceptual schema instance $J_{T'}$		
HardWorker(1, John)	Worker(1, John)	WorksIn(1, Marketing)
HardWorker(2, Luke)	Worker(2, Luke)	WorksIn(2, Shipping)
LazyWorker(3, Mike)	Worker(3, Mike)	WorksIn(3, Shipping)
Department(Marketing)	HasManager(Marketing, 1)	
Department(Shipping)	HasManager(Shipping, 2)	
c. Target instance J_T		
$Employee(1, John, Mark.)$	$HasPassedTest(1, N_1)$	$Dept(Mark.)$
$Employee(2, Luke, Ship.)$	$HasPassedTest(2, N_2)$	$Dept(Ship.)$
$Employee(3, Mike, Ship.)$		$Manager(1, Mark.)$
		$Manager(2, Ship.)$

Fig. 4. Source, conceptual, and target instances

To formalize this notion, given a source-to-semantic scenario $\mathcal{M}_{ST'} = \{\mathbf{S}, \mathbf{T}', \Sigma_{ST'}, \Sigma_{T'}\}$ with view definitions Σ_V , we say that the source-to-target rewritten scenario $\mathcal{M}_{ST} = \{\mathbf{S}, \mathbf{T}, \Sigma_{ST}, \Sigma_T\}$ is *correct* if, for each instance I of the source database, whenever a universal solution J_T for I and \mathcal{M}_{ST} exist, then $\Sigma_V(J_T)$ is also a universal solution for I and the original scenario $\mathcal{M}_{ST'}$.

Figure 4.c reports a correct target solution for I (N_1, N_2 are labeled nulls). It can be seen, in fact, that $\Sigma_V(J_T)$ is exactly the conceptual instance $J_{T'}$ in Figure 4.b. Note that a different font is used for entity and relationship types in the conceptual instance.

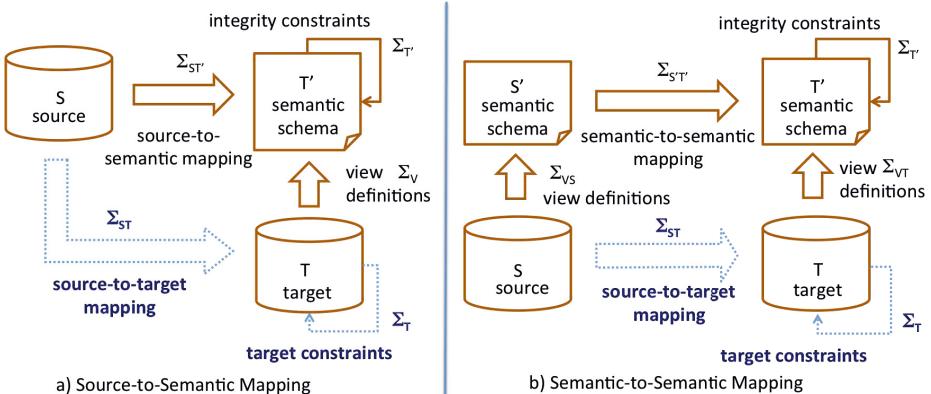


Fig. 5. Semantic Mapping Scenarios

Semantic-to-Semantic Mappings. The following sections are devoted to the development of the mapping rewriting algorithm. Before we turn to that, let us discuss what happens when also a conceptual schema over the source is given, as shown in 5.b. In this case, we assume that both view definitions for the source conceptual schema, Σ_{VS} , and target conceptual schema, Σ_{VT} are given, with the

respective egds. We also assume that the mapping, $\Sigma_{S'T'}$, is designed between the two conceptual descriptions.

It can be seen that this case can be reduced to the one above. In fact, we can see the problem as the composition of two steps: (i) applying the source view definitions to materialize the extent of the source conceptual schema; (b) consider the materialized instance as a new source database, and solve the source-to-semantic mapping problem as in Figure 5.a. In light of this, in the following we shall concentrate on the scenario in Figure 5.a only.

4 The Rewriting Algorithm

Given our input source-to-semantic mapping scenario, $\mathcal{M}_{ST'} = \{\mathbf{S}, \mathbf{T}', \Sigma_{ST'}, \Sigma_{T'}\}$, our approach consists in progressively rewriting dependencies in $\Sigma_{ST'}$ and $\Sigma_{T'}$ in order to remove view symbols, and replace them with target relations. To do this, we shall apply a number of transformations that guarantee that the rewritten mapping yields equivalent results wrt to input one, in the sense discussed in Section 3.

Normalization. The first of these transformations consists in *normalizing* the tgds in $\Sigma_{ST'}$. A tgd is said to be in *normal form* [13] if it cannot be split in two or more equivalent tgds with the same premise. In our example, the tgds in Figure 3 are not normalized. To give an example, consider m_2 :

$$m_2 : \text{Dept}(name, mgr) \rightarrow \exists mname : \text{Department}(name), \text{HasManager}(name, mgr), \\ \text{HardWorker}(mgr, mname), \text{Worker}(mgr, mname)$$

The tgd can be split in several tgds in normal form. The intuition behind the normalization is to break atoms in the conclusion in such a way that two atoms remain together only if they share an existential variable, as follows:

$$m_{2a} : \text{Dept}(name, mgr) \rightarrow \text{Department}(name) \\ m_{2b} : \text{Dept}(name, mgr) \rightarrow \text{HasManager}(name, mgr) \\ m_{2c} : \text{Dept}(name, mgr) \rightarrow \exists mname : \text{HardWorker}(mgr, mname), \text{Worker}(mgr, mname)$$

Similarly, we can split m_0 into four tgds, $m_{0a}, m_{0b}, m_{0c}, m_{0d}$, and m_1 into $m_{1a}, m_{1b}, m_{1c}, m_{1d}$, each with a single atom in the conclusion. In the following, we shall always refer to these normalized tgds.

Unfolding the View Definition Language. Once tgds have been normalized, we may proceed with the goal of removing view symbols from mapping conclusions. It can be seen that our problem reduces to a variant of the classical problem of *view unfolding*.

As it is obvious, the complexity of the problem depends quite a lot on the expressibility of the view definition language allowed in our scenarios. In fact, if we used plain conjunctive queries over target relation symbols as a view definition language, the rewriting would pretty much reduce to an application of the standard view unfolding algorithm [22]. To give an example, consider mapping m_{0c} and recall the definition of view `WorksIn`:

$$m_{0c} : \text{Emp}(id, name, dept, rating) \rightarrow \text{WorksIn}(id, dept) \\ \text{WorksIn}(id, dept) \Leftarrow \text{Employee}(id, name, dept)$$

Standard view unfolding yields the following s-t tgd:

$$m'_{0c} : \text{Emp}(id, name, dept, rating) \rightarrow \exists name' : \text{Employee}(id, name', dept)$$

However, the main purpose of having a semantic description of the target database stands in its richer nature with respect to the power of the pure selection-projection-join paradigm. In fact, in this paper we allow for a more expressive language than conjunctive queries, i.e., non-recursive Datalog with negation.

Since the standard view-unfolding algorithm cannot handle negated atoms, we need to devise a new and improved algorithm to correctly unfold non-recursive Datalog view definitions with negations. However, before doing that we need to ask ourselves if the new view definition language allows for this kind of unfolding.

Unfortunately, we have the following negative result, stating that the semantic-based mapping problem cannot in general be solved for the full language of non-recursive Datalog with negation:

Proposition 1. *There exists a semantic schema T' defined in terms of non-recursive Datalog rules with negation and a source-to-semantic scenario $M_{ST'} = \{\mathbf{S}, \mathbf{T}', \Sigma_{ST'}, \Sigma_{T'}\}$, such that there exists no correct rewriting in terms of tgds, egds, and denials only.*

The main intuition behind the proof is that the increased expressive power of the view definition language requires a more expressive language to rewrite the tgds. This new tgd language is the language of *disjunctive embedded dependencies* [7], i.e., tgds in which disjunction symbols are allowed in the conclusion. Since computing solutions for DEDs is significantly more challenging than for standard dependencies, in this paper we rather concentrate on standard tgds and egds.

Proposition 1 leaves us with a crucial question: is it possible to find a view definition language that is at the same time more expressive than plain conjunctive queries, and allows in practice to compute correct rewritings in terms of tgds and egds?

In the following sections, we show that this language exists, and corresponds to non-recursive Datalog with a limited negation. To be more precise, we limit negation in such a way that: (i) only one negated atom is allowed for each view definition; (ii) keys and functional dependencies – i.e., egds – are defined only for views whose definition does not depend on negated atoms.

In the rest of the paper, we concentrate exactly on this language.

The View Unfolding Algorithm. The pseudocode of our unfolding algorithm *UnfoldDependencies* is reported in Algorithm 1. The algorithm takes as input the initial set of source-to-semantic dependencies, $\Sigma_{ST'}$, and the semantic egds in $\Sigma_{T'}$, plus the view definitions in Σ_V , and returns a set of source-to-target tgds, Σ_{ST} , plus a set of target egds, tgds and denials Σ_T . To define the algorithm, we use the standard unfolding algorithm for positive views, *unfoldView* [22], as a building block.

The main intuition behind the algorithm is easily stated: it works with a set of dependencies, called Σ , initialized as $\Sigma_{ST'} \cup \Sigma_{T'}$, and progressively transforms this set, until a fixpoint is reached. Note that it always terminates, since we

Algorithm 1. *UnfoldDependencies*($\Sigma_{ST'}$, $\Sigma_{T'}$, Σ_V)

```

 $\Sigma := \Sigma_{ST'} \cup \Sigma_{T'}$ 
repeat
  for all  $d \in \Sigma$  do
    if  $d$  contains a positive derived atom  $L$  then
       $d := unfoldView(L, d, \Sigma_V)$ 
    end if
    if  $d$  is a tgd containing a negative derived atom  $\neg L(\bar{x}_i)$  in  $\psi(\bar{x}, \bar{y})$  then
       $d := \phi(\bar{x}) \rightarrow \psi(\bar{x}, \bar{y}) - \{\neg L(\bar{x}_i)\}$ 
      let  $TGD_i$  be a new relation symbol
       $d^1 := \phi(\bar{x}) \rightarrow TGD_i(\bar{x})$ 
       $d^2 := TGD_i(\bar{x}) \wedge L(\bar{x}_i) \rightarrow \perp$ 
       $\Sigma := \Sigma \cup \{d^1, d^2\}$ 
    end if
    if  $d$  contains a negative atom  $\neg L(\bar{x}_i)$  in  $\phi(\bar{x})$  then
       $d := \phi(\bar{x}) - \{\neg L(\bar{x}_i)\} \rightarrow \psi(\bar{x}, \bar{y}) \cup \{L(\bar{x}_i)\}$ 
    end if
  end for
until fixpoint
 $\Sigma_{ST} :=$  the set of s-t tgds in  $\Sigma$ 
 $\Sigma_T :=$  the set of egds, target tgds and denials in  $\Sigma$ 

```

assume the view definitions are not recursive. Three main transformations are employed in order to remove derived atoms from the dependencies of Σ :
(i) first, whenever a positive derived atom $L(\bar{x}_i)$ is found in a dependency d , the algorithm uses the standard view unfolding algorithm as a building block in order to replace $L(\bar{x}_i)$ by its view definition; to see an example, consider tgd m_0 and view `LazyWorker`:

$$\begin{aligned} m_{0a} : & Emp(id, name, dept, rating), rating < 5 \rightarrow \text{LazyWorker}(id, name) \\ & \text{LazyWorker}(id, name) \Leftarrow Employee(id, name, dept), \neg \text{Promoted}(id) \end{aligned}$$

Standard unfolding changes m_0 as follows:

$$m'_{0a} : Emp(id, name, dept, rating), rating < 5 \rightarrow \exists \text{dept}' : Employee(id, name, dept'), \neg \text{Promoted}(id)$$

(ii) the second, and most important transformation, consists in handling negated view atoms $\neg L(\bar{x}_i)$ in tgd conclusions, as the one about `Promoted` employees in m'_{0a} ; we cannot directly unfold the atom in the conclusion in order to have an equivalent tgd; we need a way to express more appropriately the intended semantics, i.e, the fact that the tgd should be fired only if it is not possible to satisfy $L(\bar{x}_i)$; to express this, we remove the negated atom from the conclusion

$$m''_{0a} : Emp(id, name, dept, rating), rating < 5 \rightarrow \exists \text{dept}' : Employee(id, name, dept')$$

and introduce two new dependencies;

– the first one, d^1 , uses a new relation symbol, $TGD_i(\bar{x})$, to express the fact that the body of d is fired for a vector of variables \bar{x} ; in our example, we need to add this new tgd:

$$m^1_{0a} : Emp(id, name, dept, rating), rating < 5 \rightarrow TGD_0(id, name, dept, rating)$$

– the second one, d^2 , states that d should fire only if it is not possible to satisfy $L(\bar{x}_i)$, by means of a denial constraint, as follows:

$$m_{0a}^2 : TGD_0(id, name, dept, rating), \text{Promoted}(id) \rightarrow \perp$$

(iii) the third, and final transformation, consists of moving negated view atoms of the form $\neg L(\bar{x}_i)$ in the premise of a dependency d to its conclusion, in order to remove the negation. To see an example of this, let's complete the rewriting of m_{0a}^2 ; transformation (i) needs to be applied again in order to unfold the Promoted atom:

$$\begin{aligned} m_{0a}^{2'} : & TGD_0(id, name, dept, rating), \text{HasPassedTest}(id, tc), \\ & \neg \text{DisciplinaryProcedure}(pnum, id) \rightarrow \perp \end{aligned}$$

however, the negative atom may be easily moved to the conclusion, to yield the final target tgd:

$$\begin{aligned} m_0^{2''} : & TGD_0(id, name, dept, rating), \text{HasPassedTest}(id, tc) \\ & \rightarrow \exists pnum : \text{DisciplinaryProcedure}(pnum, id) \end{aligned}$$

It is worth discussing why this last step is safe in our setting. Notice, in fact, that while moving a single negated atom from the premise to the conclusion gives a correct rewriting, if the atoms were more than one a disjunction would be needed in the conclusion; in this way, we would end up again with disjunctive embedded dependencies, and therefore go outside the domain of standard dependencies.

However, given the restrictions that we have imposed on our view definition language, this cannot be the case; notice in fact that (a) negated view atoms may appear only in the premise of denials introduced by transformation; the initial tgds only contain source symbols in the premise, and we make the assumption that no egd is defined over a view whose definition involves negation; (b) by construction, these denials are such that only one negation may appear in the body; in fact, for each view symbol in a tgd conclusion only one negation may appear.

We are now ready to state our main result, about the correctness of the rewriting algorithm. Before we do that, we should make it more precise the schemas that are involved in the translation. In fact, we start with a target schema, \mathbf{T} , but during the rewriting we enrich it with new relation symbols, TGD_0, TGD_1, \dots , in order to be able to correctly specify denials. We shall call the resulting schema \mathbf{T}'' .

Theorem 1 (Correctness). *Given a source-to-semantic scenario $\mathcal{M}_{ST'} = \{\mathbf{S}, \mathbf{T}', \Sigma_{ST'}, \Sigma_{T'}\}$ with view definition Σ_V , algorithm UnfoldDependencies computes a correct source-to-target rewritten scenario $\mathcal{M}_{ST''} = \{\mathbf{S}, \mathbf{T}'', \Sigma_{ST''}, \Sigma_{T''}\}$, where \mathbf{T}'' is obtained from \mathbf{T} by enriching it with a finite set of new relation symbols TGD_0, TGD_1, \dots*

5 Experiments

We have implemented a chase engine and a prototype of the algorithm presented in this paper, both written in Java. The experiments have been performed on

an Intel core i7 machine with a 2.6 GHz processor, 8 GB of RAM, and running MacOSX. The DBMS used has been PostgreSQL 9.2.1 (x64 version).

Datasets. We used two datasets. The first one (**EMPLOYEES**), correspond to our running example; of this, we have studied two variants, one with egds, the other with no egds. The second one (**SYNTHETIC**), is a fully synthetic dataset of which we generated variants from 50 to 40K dependencies.

Effectiveness. As a first set of experiments, we study the effectiveness of our approach. More specifically, we compared the size of the source-to-semantic mapping that users need to specify for the various scenarios, to the size of the actual source-to-target scenario generated by our rewriting. As a measure of the size of a scenario, we took the number of nodes and edges of the *dependency graph* [8], i.e., the graph in which each atom of a dependency is a node, and there is an edge from node n_1 to node n_2 whenever the corresponding atoms share a variable. Intuitively, the higher is the complexity of this graph, the more complicated is to express the mapping. Figure 6.a reports the results for 4 scenarios. It can be seen that in all scenarios there was a considerable increase in the size of the dependency graph (up to 70%). This is a clear indication that in many cases our approach is more effective with respect to manually developing the source-to-target mapping.

Scalability on Large Scenarios. The second set of experiments tests the scalability of our unfolding algorithm on mapping scenarios of large size. The results of these experiments are reported on Figure 6.b. We have generated a series of synthetic scenarios of increasing size. All source-to-semantic tgds in these scenarios have two source relations on the premise and two views on the conclusion. Each view definition has two positive target relations and (at most) one negative auxiliary view. For each mapping scenario, 20% of the tgds have no negated atoms, the next 20% have 1 level of negation (i.e., negated atoms that do not depend in turn on other negations), the next 20% have 2 levels of negation, and so on, up to 4 levels of negations. The number of source relations in the mapping scenarios ranges from 10k to 80k, the number of view definitions ranges from 30k to 240k, and the number of target relations ranges from 60k to 480k. The reported times are the running times of the unfolding algorithm running in main memory, and do not include disk read and write times. As it can be seen, the rewriting algorithm scales nicely to large scenarios.

Scalability on Large Datasets. The final set of experiments tests the scalability of the chase engine on large databases. This is a very important issue, since previous works [14,12] have shown that existing chase engines for egds may require several hours on a few thousands tuples, and therefore hardly scale to large datasets. Figure 6.c reports the time needed to compute a solution for three of our scenarios. As it was expected, scenarios with no egds required lower computing times. However, also in the case of egds, our chase engine scaled nicely to databases of 1 million tuples. To the best of our knowledge, this is the first actual scalability result for the chase of egds.

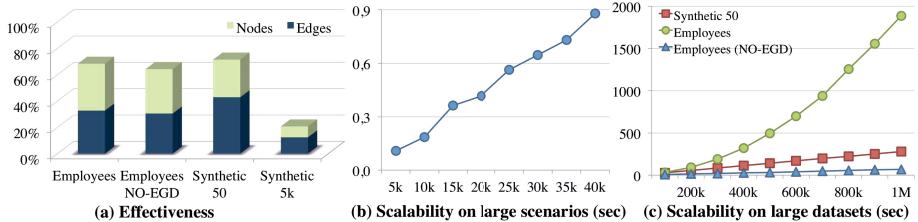


Fig. 6. Results of Experiments

6 Related Works

The standard view unfolding algorithm [22] has been extensively used in data integration as a tool for query answering. In such a setting, users pose queries over a set of heterogeneous sources through a single global schema, which provides a uniform view of all the sources. Mappings between the sources and the global schema are used to rewrite the users' queries in terms of the sources. One way to define these mappings is the so-called global-as-view approach (GAV), in which the global schema is defined as a view over the sources. With this kind of mappings, answering a query posed on the global schema usually reduces to unfolding the view definitions [11] (unless integrity constraints are present in the global schema, which makes answering harder [4]).

Another similar problem is that of accessing data through ontologies, in which users pose queries on an ontology that is defined on top of a set of databases; the ontology plays the role of global schema, and the databases play the role of data sources [18,5]. The problem we address in this paper, however, is not about using view unfolding to answer queries, but to copy data into a target. As we have discussed in Section 4, standard view unfolding suffices only when the views that define the target conceptual schema in terms of the underlying database are plain conjunctive queries. In the presence of negation, copying data into the target gets more complicated, as negated atoms in mapping conclusions introduce new integrity constraints that standard view unfolding does not handle (intuitively, negated atoms must be kept false during all the process of copying data into the target).

A problem that relates to our use of view unfolding in mappings is that of mapping composition [10,16]. Composing a mapping between schemas **A** and **B** with a mapping between schemas **B** and **C** produces a new mapping between **A** and **C**. In a sense, our application of view unfolding to the conclusion of a mapping can be seen as a kind of mapping composition; one in which the mapping between the source and the conceptual schema is composed with a second mapping that relates the conceptual schema with the underlying database (i.e., the views). However, mapping composition techniques take into account the direction of the mapping, that is, one can compose a mapping from **A** to **B** only with another mapping that goes from **B** to some **C** in order to get a mapping that goes from **A** to **C**. In our case, we have a mapping from the source to the

conceptual schema and another one from the database to the conceptual schema, which cannot be directly composed.

The introduction of conceptual schemas into the mapping process has also been investigated in [1] with respect to a different problem, i.e., that of generating mappings between databases. Since we assume that source-to-semantic mappings are given as inputs, the techniques developed in [1] can be used as a preliminary step to simplify the mapping specification phase.

Another context where mappings involving conceptual schemas have been studied is that of semantic-web ontologies; in particular, [21] proposes a technique that translates a set of correspondences between a source and target ontologies into a set of SPARQL queries that can then be run against the data source to produce the target's data. Comparing with our approach, we assume that the given mapping is not just a set of correspondences, but a complete declarative mapping expressed as tgds, and we also take into account that the target's conceptual schema is a view of the underlying database.

Mappings between conceptual schemas have also been studied in [2], where the authors propose an approach for finding "semantically similar" associations between two conceptual schemas. This similar associations are then used to generate a mapping. This approach is complementary to ours in the sense that it could be used to generate a semantic-based mapping, which would then be rewritten using the algorithm we present in this paper.

References

1. An, Y., Borgida, A., Miller, R., Mylopoulos, J.: A Semantic Approach to Discovering Schema Mapping Expressions. In: ICDE, pp. 206–215 (2007)
2. An, Y., Song, I.-Y.: Discovering semantically similar associations (sesa) for complex mappings between conceptual models. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 369–382. Springer, Heidelberg (2008)
3. Beeri, C., Vardi, M.: A Proof Procedure for Data Dependencies. J.ACM (1984)
4. Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M.: Data integration under integrity constraints. Inf. Syst. 29(2), 147–163 (2004)
5. Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: View-based query answering in description logics: Semantics and complexity. J. Comput. Syst. Sci. 78(1), 26–46 (2012)
6. Ceri, S., Gottlob, G., Tanca, L.: What you Always Wanted to Know About Datalog (And Never Dared to Ask). IEEE TKDE 1(1), 146–166 (1989)
7. Deutsch, A., Tannen, V.: Optimization properties for classes of conjunctive regular path queries. In: Ghelli, G., Grahne, G. (eds.) DBPL 2001. LNCS, vol. 2397, pp. 21–39. Springer, Heidelberg (2002)
8. Fagin, R., Kolaitis, P., Miller, R., Popa, L.: Data Exchange: Semantics and Query Answering. TCS 336(1), 89–124 (2005)
9. Fagin, R., Kolaitis, P., Nash, A., Popa, L.: Towards a Theory of Schema-Mapping Optimization. In: PODS, pp. 33–42 (2008)
10. Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.C.: Composing schema mappings: Second-order dependencies to the rescue. ACM TODS 30(4), 994–1055 (2005)
11. Lenzerini, M.: Data integration: a Theoretical Perspective. In: PODS (2002)

12. Marnette, B., Mecca, G., Papotti, P.: Scalable data exchange with functional dependencies. *VLDB* 3(1), 105–116 (2010)
13. Mecca, G., Papotti, P., Raunich, S.: Core Schema Mappings. In: SIGMOD (2009)
14. Mecca, G., Papotti, P., Raunich, S.: Core Schema Mappings: Scalable Core Computations in Data Exchange. *Inf. Syst.* 37(7), 677–711 (2012)
15. Miller, R.J., Haas, L.M., Hernandez, M.A.: Schema Mapping as Query Discovery. In: VLDB, pp. 77–99 (2000)
16. Nash, A., Bernstein, P.A., Melnik, S.: Composition of mappings given by embedded dependencies. *ACM Trans. Database Syst.* 32(1), 4 (2007)
17. Olivé, A.: Conceptual modeling of information systems. Springer (2007)
18. Poggi, A., Lembo, D., Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: Linking data to ontologies. *J. Data Semantics* 10, 133–173 (2008)
19. Popa, L., Velegrakis, Y., Miller, R.J., Hernandez, M.A., Fagin, R.: Translating Web Data. In: VLDB, pp. 598–609 (2002)
20. Queralt, A., Teniente, E.: Reasoning on UML class diagrams with OCL constraints. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 497–512. Springer, Heidelberg (2006)
21. Rivero, C.R., Hernández, I., Ruiz, D., Corchuelo, R.: Generating SPARQL executable mappings to integrate ontologies. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 118–131. Springer, Heidelberg (2011)
22. Sterling, L., Shapiro, E.Y.: The Art of Prolog: Advanced Programming Techniques. MIT Press (1994)
23. ten Cate, B., Chiticariu, L., Kolaitis, P., Tan, W.C.: Laconic Schema Mappings: Computing Core Universal Solutions by Means of SQL Queries. *VLDB* 2(1), 1006–1017 (2009)

Managing Security Requirements Conflicts in Socio-Technical Systems

Elda Paja¹, Fabiano Dalpiaz², and Paolo Giorgini¹

¹ University of Trento, Italy

{elda.paja, paolo.giorgini}@unitn.it

² University of Toronto, Canada

dalpiaz@cs.toronto.edu

Abstract. Requirements are inherently prone to conflicts, for they originate from stakeholders with different, often opposite, needs. Security requirements are no exception. Importantly, their violation leads to severe effects, including privacy infringement, legal sanctions, and exposure to security attacks. Today’s systems are Socio-Technical Systems (STSs): they consist of autonomous participants (humans, organisations, software) that *interact* to get things done. In STSs, security is not just a technical challenge, but it needs to consider the social components of STSs too. We have previously proposed STS-ml, a security requirements modelling language for STSs that expresses security requirements as contractual constraints over the interactions among STS participants. In this paper, we build on top of STS-ml and propose a framework that, via automated reasoning techniques, supports the identification and management of conflicts in security requirements models. We apply our framework to a case study about e-Government, and report on promising scalability results of our implementation.

Keywords: security requirements, automated reasoning, requirements models.

1 Introduction

Socio-Technical Systems (STSs) are complex systems composed of autonomous subsystems (*participants*), which are either technical (software) or social (humans and organisations). These subsystems interact to achieve objectives they cannot achieve on their own and to exchange information. STSs are loosely controllable, for their participants are autonomous. Autonomy makes the design of secure STSs a challenging task. For example, if a participant transfers confidential information to another, how does she know that data is kept confidential, without having control?

Goal-oriented approaches to security requirements engineering [7,12,13] offer a suitable abstraction level for the design of secure STSs. They model STSs as a set of actors that are *intentional*—they have objectives—and *social*—they interact with others to achieve their objectives. Unfortunately, their underlying ontology is too abstract to effectively represent real-world information security requirements, which include fine-grained and contradictory authorisations over information entities [1,19].

To overcome this limitation, we have previously proposed the Socio-Technical Security modelling language for STSs (STS-ml) [2]. The language relies on a more expressive ontology and its security requirements are relationships between couples of STS

actors, where a *requester* actor requires a *requestee* actor to comply with a security need. Each participant expresses her own requirements. STS-ml models are actor- and goal-oriented, and they represent the business policies of the participants, their security requirements over information and goals, and organisational constraints.

Being specified independently by different actors, policies and security requirements are likely to clash, thus leading to inconsistent specifications that cannot be satisfied by an implemented STS (at least one requirement would be violated). The detection and handling of conflicts between requirements is a hard task [5] (goal-models tend to become huge and complex), and it often requires the usage of automated reasoning techniques. This is true in STS-ml too, for the language supports complex security requirements (due to its expressiveness) and real-world models are typically large [17].

In this paper, we propose a framework for managing conflicts in STS-ml requirements. Our framework suggests to iteratively (i) create STS-ml models for the domain at-hand, (ii) identify conflicts through automated reasoning techniques, and (iii) resolve conflicts. Specifically, we focus here on the first two steps, and leave conflicts resolution for future work. We address two types of conflicts: among security requirements, and between actors' business policies and security requirements. We consider the interplay between different requirements sources: the business policies of individual actors, their security expectations on other actors, and the organisational constraints in the STS.

The contributions of the paper are as follows:

- A revised version of the STS-ml modelling language that includes a richer set of security requirements along with their formal definition.
- A framework for detecting conflicts between (i) security requirements, as well as (ii) among participants' business policies and security requirements.
- An implementation of our framework—using disjunctive Datalog logic programs—as essential part of a CASE tool called STS-Tool.
- Results from a case study, which show the effectiveness of our framework in identifying non-trivial conflicts as well as promising scalability results.

Section 2 reviews related work. Section 3 presents a case study about e-Government. Section 4 presents the current version of STS-ml and its supported security requirements. Section 5 describes our framework for identifying conflicts, while Section 6 evaluates it on the case study and reports on scalability results. Finally, Section 7 presents conclusions and future directions.

2 Related Work

We review related work concerning identifying conflicting requirements, reasoning about security requirements, and methodologies for security requirements engineering.

Conflicting Requirements. The importance of handling conflicts in requirements is well-known in practice and has been acknowledged by the research community [6,18]. We review here the main frameworks in goal-oriented requirements engineering.

Giorgini et al. [8] analyse goal satisfaction/denial by mapping goal models to the satisfiability problem. Their analysis determines evidence of goal satisfaction/denial by using label propagation algorithms. Conflicts occur in case of both positive and

negative evidence. Their approach inspired further research. Horkoff and Yu [10] deal with conflicts by demanding resolution to the analyst in an interactive fashion. Fuxman et al. [6] use first-order linear-time temporal logic to identify scenarios with conflicts. KAOS [18] includes analysis techniques to identify and resolve inconsistencies that arise from the elicitation of requirements from stakeholders with different viewpoints.

All these approaches detect conflicting goals. Our approach, instead, treats security requirements as relationships between actors. These approaches could be used to detect inconsistencies within an individual business policy, i.e., within the scope of one actor.

Reasoning about Security Requirements. SI* [7] is a security requirements engineering framework that builds on *i** [20] by adding security concepts, including delegation and trust of execution or permission. SI* uses automated reasoning to check security properties of a model, reasoning on the interplay between execution and permission of trust and delegation relationships. STS-ml supports a more expressive ontology (featuring fine-grained authorisations) to represent information security requirements, and decouples business policies (an actor's goals) from security requirements.

De Landtsheer and van Lamsweerde [3] model confidentiality claims as specification patterns, representing properties that unauthorised agents should not know. They identify violations of confidentiality claims in terms of counterexample scenarios present in requirements models. While their approach represents confidentiality claims in terms of high-level goals, ours represents authorisation requirements as social relationships, and we identify violations by looking at the business policies of the actors.

Security Requirements Methodologies. These are methods to identify possible conflicts, as opposed to using automated reasoning. Secure Tropos [13] models security concerns throughout the whole development process. It expresses security requirements as *security constraints*, considers potential threats and attacks, and provides methodological steps to validate these requirements and overcome vulnerabilities.

Liu et al. [12] extend *i** to deal with security and privacy requirements. Their methodology defines security and privacy-specific analysis mechanisms to identify potential attackers, derive threats and vulnerabilities, thereby suggesting countermeasures.

Haley et al. [9] construct the context of the system, define security requirements as constraints over functional requirements, and develop a structure of satisfaction arguments to verify the correctness of security requirements. This approach focuses mainly on system requirements, while ours is centred on the interaction among actors.

3 Motivating Case Study: Tax Collection in Trentino

Trentino as a Lab (TasLab)¹ is an online collaborative platform to foster ICT innovation among research institutions, universities, enterprises, and public administration in the Trentino province [16]. We focus on a TasLab collaborative project concerning tax collection. The Province of Trento (*PAT*) and the Trentino *Tax Agency* require a system that verifies if correct revenues are collected from *Citizens* and *Organisations*, provides a complete profile of taxpayers, generates reports, and enables online tax payments.

¹ <http://www.taslab.eu>

This is an STS in which actors interact via a technical system: citizens and organisations pay taxes online; municipalities (*Municipality*) furnish information about citizens' tax payments; Informatica Trentina (*InfoTN*) is the system contractor; other IT companies develop specific functionalities (e.g., data polishing); the Tax Agency is the system end user; and PAT withholds the land register (information about buildings and lots).

These actors exchange documents that contain confidential information. Each actor has a business policy for achieving her goals, and expects others to comply with her security requirements (e.g., integrity and confidentiality). Organisational constraints (rules and regulations) apply to all actors. Different types of conflict may arise:

- Business policies can clash with security requirements. The Tax Agency's security requirements may include authorising InfoTN to read some data, but they prohibit further transmission of such data. If InfoTN's business policy includes relying upon an external provider to polish data, a conflict would occur.
- Security requirements can be conflicting. For instance, citizens' security requirements may include prohibiting IT companies to access their personal data, while the municipality's (possessing citizens' records) security requirements towards IT companies may specify granting them such authority.
- Organisational constraints may make some requirements unsatisfiable. For instance, a constraint may prohibit private subjects from matching citizens' personal information with tax records. This could create a conflict with the business policy of the data polishing company, which specifies as necessary matching such information.

4 The STS-ml Framework for Security Requirements Modelling

STS-ml is a security requirements modelling framework for designing secure STSs [2]. We present a revised version of STS-ml, which introduces (i) a specialised set of security requirements over information including non-reading, non-modification, non-production, non-disclosure, and non-reauthorisation; (ii) two specialisations of the non-repudiation security requirement; (iii) four specialisations of redundancy; and (iv) requirements over the exchange of documents, such as integrity of transmission; and (v) security requirements from rules and regulations, such as separation and binding of duties.

In this section, we formalise the modelling primitives of STS-ml (Section 4.1) and the security requirements that STS-ml supports (Section 4.2).

4.1 Modelling with STS-ml

STS-ml models are constructed by iteratively building three views (*social*, *information*, and *authorisation*), each focussing on different aspects of the STS. An STS-ml model consists of all the elements and relationships from the three views. The multi-view modelling feature of STS-ml promotes modularity and separation of concerns. Figure 1 shows part of the model for our case study (the complete model is available in [14]).

Social View. This represents actors as intentional (having goals that they want to attain) and social (interacting via goal delegations and document exchange) entities. STS-ml supports two types of actors: *agents* are concrete participants, while *roles* are abstract actors, used when the actual participant is unknown. We model InfoTN as an agent,

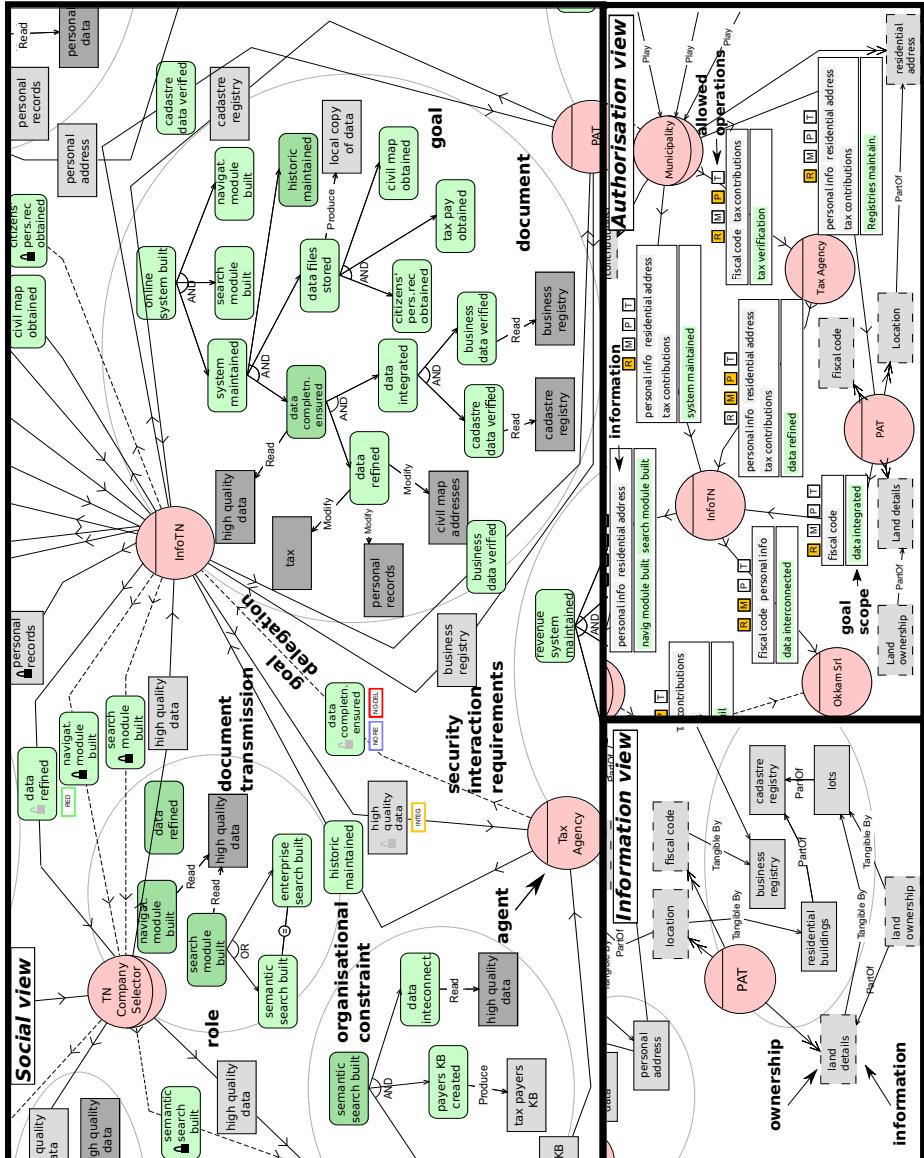


Fig. 1. Partial STS-ml model of the tax collection case study

while Municipality is a role that can be adopted by any municipality in the province (Figure 1). InfoTN has goals online system built, data completn. ensured, and so on. Actors refine their goals through and/or decompositions: InfoTN and-decomposes goal data completn. ensured into data refined and data integrated. Actors may possess documents; they may read, modify, or produce documents while achieving their goals. InfoTN modifies tax for data refined. Actors can transmit documents to others only if

they possess the required document. PAT possesses the business registry and transmits it to InfoTN, which reads this document in order to have business data verified.

An actor's **business policy** defines alternative strategies for an actor to achieve her root goals. It is a sub-model of the social view that includes all the goals and documents in the scope of that actor in the social view, the relationships (and/or-decompose, read, modify, and produce) among those goals and documents, as well as goal delegations and document transmissions that start from that actor and end to another actor.

The business policy of TN Company Selector includes goals data refined, navigation module built, for which document high quality data is read, and search module built that the actor or-decomposes into semantic search built and enterprise search built (Figure 1). These subgoals denote alternative strategies: the actor can choose either of them.

Information View. This defines the relation between information (the data that actors care of) and documents (artifacts that represent information), and represents information ownership. Information can be represented by one or more documents (Tangible By). One or more information pieces can be part of some document. The linkage between information and documents is key to identify which information actors affect, while *reading*, *modifying*, *producing*, or *transmitting* documents. In Figure 1, information on fiscal code owned by PAT is made tangible by document business registry. Also, the information view structures information and documents through part-of relations. For instance, residential buildings is part of cadastre registry.

Authorisation View. This defines the requirements of the actors about who can access information, for what purpose, and which operations can be performed on documents representing information. Authorisations are granted on a set of information \mathcal{I} , specifying the operations the authorisee is allowed to perform from the set $\{R, M, P, T\}$ (where R-read, M-modify, P-produce, T-transmit), and the purpose (which goals) \mathcal{G} for which the authorisation is passed. If $\mathcal{G} = \emptyset$, no restriction about the purpose exists. STS-ml authorisations also specify if *authority to delegate* authority to others is granted (transferrable). The Municipality authorises InfoTN to read (R) information personal info, residential address, and tax contributions, in the scope of goal system maintained granting authority to delegate (graphically, the authorisation has a continuous arrow line).

4.2 Security Requirements in STS-ml

STS-ml aims to ensure information security and secure interaction among participants. The supported requirements originate from the main security aspects in the NIST glossary [11], they have been refined through collaboration with industry², and confirmed by evaluation workshops [17]. STS-ml supports three types of security requirements:

- *Interaction (security) requirements* are security-related constraints actors express over their interactions, namely goal delegations and document transmissions.
- *Organisational constraints* determine another range of requirements that constrain the adoption of roles and the uptake of responsibilities.
- *Authorisation requirements* determine authorisations and prohibitions over information specifying whether they can be used, how, for which purpose, and by whom.

² Our partners in the EU FP7 Project Aniketos <http://www.aniketos.eu>

An STS-ml model is *consistent* when there are no conflicts between security requirements and business policies.

Interaction (Security) Requirements. STS-ml supports several interaction security requirements on goal delegations and document transmissions (illustrated in Figure 1). Let Del stand for delegate(A_1, A_2, G), where A_1 and A_2 are actors, G is a goal:

- *Non-repudiation of delegation/acceptance* [$R_1: r\text{-not-repudiated-del}(A_2, A_1, Del)$, $R_2: r\text{-not-repudiated-acc}(A_1, A_2, Del)$]: A_2 (A_1) requires A_1 (A_2) not to repudiate the delegation (acceptance of the delegation) Del . In Figure 1, Tax Agency wants InfoTN not to repudiate the acceptance of delegation for data completeness ensured.
- *Redundancy* [$r\text{-red-ensured}$]: the delegatee has to adopt redundant strategies for the achievement of a delegated goal, either by (a) relying on other actors, or by (b) using alternative internal capabilities. We consider two types of redundancy: (1) *Fallback redundancy*: a primary strategy is selected to fulfill the goal, while other strategies are maintained as backup, and are used only if the primary strategy fails. (2) *True redundancy*: two or more different strategies are executed simultaneously. By intertwining (a-b) with (1-2), STS-ml supports four mutually exclusive redundancy security requirements: (a1) true redundancy single [$R_3: r\text{-ts-red-ensured}(A_1, A_2, G)$], (a2) fallback redundancy single [$R_4: r\text{-fs-red-ensured}(A_1, A_2, G)$], (b1) true redundancy multi [$R_5: r\text{-tm-red-ensured}(A_1, A_2, G)$], and (b2) fallback redundancy multi [$R_6: r\text{-fm-red-ensured}(A_1, A_2, G)$]. For instance, InfoTN requires TN Company Selector to ensure true redundancy single for goal data refined.
- *Not-redelegation* [$R_7: r\text{-not-redelegated}(A_1, A_2, G)$]: the delegator wants the delegatee not to further delegate goal fulfilment. InfoTN requires TN Company Selector not to redelegate goal search module built, for instance.

Let Tx stand for transmit(A_1, A_2, Doc), where A_1, A_2 are actors, Doc is a document:

- *Integrity of transmission* [$R_8: r\text{-integrity-ensured}(A_2, A_1, Tx)$]: requires the sender to guarantee the integrity of Doc while transmitting it. Tax Agency requires InfoTN to guarantee the transmission integrity of high quality data.

Organisational Constraints. These requirements do originate from laws, business rules, and regulations. In these constraints, the *STS* is the legal regulatory context for the considered domain. STS-ml supports two basic types of organisational constraints: *Separation of Duties* (SoD), and *Binding of Duties* (BoD). These constraints dictate restrictions over role-to-role relationships as well as agent-to-role and goal-to-goal relationships:

- *Role-based SoD* [$R_9: r\text{-not-played-both}(STS, A, R_1, R_2)$]: defines that two roles are incompatible, i.e., no agent A can play both roles R_1 and R_2 .
- *Role-based BoD* [$R_{10}: r\text{-played-both}(STS, A, R_1, R_2)$]: defines a binding between roles, i.e., if agent A plays role R_1 (R_2), then A must also play R_2 (R_1).
- *Goal-based SoD* [$R_{11}: r\text{-not-pursued-both}(STS, A, G_1, G_2)$]: defines incompatible goals, i.e., every agent A must not pursue both goals G_1 and G_2 .
- *Goal-based BoD* [$R_{12}: r\text{-pursued-both}(STS, A, G_1, G_2)$]: defines that if an agent A pursues goal G_1 (G_2), A should pursue G_2 (G_1) too. An example of this requirement is expressed between goals semantic search built and enterprise search built.

Authorisation Requirements. These are obtained from authorisations in the authorisation view. If an actor A_2 has no incoming authorisation for any information I from

\mathcal{I} , then A_2 has a prohibition for I . Such prohibition is an STS-ml authorisation from the information owner to A_2 where no operation is allowed nor authority to transfer is passed. Let $Auth$ stand for $authorise(A_1, A_2, \mathcal{I}, \mathcal{G}, \mathcal{OP}, TrAuth)$, where A_1, A_2 are actors, \mathcal{I} is a set of information, \mathcal{G} is a set of goals, \mathcal{OP} is the set of operations $\{\text{R, M, P, T}\}$, and $TrAuth$ is a boolean value determining if the authorisation is transferrable:

- $\mathcal{G} \neq \emptyset \rightarrow Need-to-know$ [R₁₃: r-not-ntk-violated($A_1, A_2, \mathcal{I}, \mathcal{G}$)]: A_2 is required not to perform any operation (read/modify/produce) on documents that make some information in \mathcal{I} tangible, for any goals not included in \mathcal{G} . The authorisation from Tax Agency to InfoTN is an example: personal info, residential address and tax contributions shall be used only for goal data refined.
- $\text{R} \notin \mathcal{OP} \rightarrow Non-read$ [R₁₄: r-not-read(A_1, A_2, \mathcal{I})] and *Not-reauthorise-read* [R₁₈: r-not-reauthorised($A_1, A_2, \mathcal{I}, \mathcal{G}, \{\text{R}\}$)]: A_2 is required not to read documents representing information in \mathcal{I} , or authorise others to do so. Tax Agency expresses such requirement on the authorisation for personal info, residential address and tax contributions to InfoTN, for authority to read those information is not granted.
- $\text{M} \notin \mathcal{OP} \rightarrow Non-modification$ [R₁₅: r-not-modified(A_1, A_2, \mathcal{I})] and *Not-reauthorise-modification* [R₁₈: r-not-reauthorised($A_1, A_2, \mathcal{I}, \mathcal{G}, \{\text{M}\}$)]: A_1 wants A_2 not to modify documents that include information in \mathcal{I} , or authorise others to do so. Municipality expresses a non-modification requirement over the authorisation towards InfoTN, when authorising it to read personal info, residential address and tax contributions.
- $\text{P} \notin \mathcal{OP} \rightarrow Non-production$ [R₁₆: r-not-produced(A_1, A_2, \mathcal{I})] and *Not-reauthorise-production* [R₁₈: r-not-reauthorised($A_1, A_2, \mathcal{I}, \mathcal{G}, \{\text{P}\}$)]: A_1 requires A_2 not to produce any documents that include information in \mathcal{I} , or authorise others to do so. For example, PAT requires InfoTN not to produce information fiscal code.
- $\text{T} \notin \mathcal{OP} \rightarrow Non-disclosure$ [R₁₇: r-not-disclosed(A_1, A_2, \mathcal{I})] and *Not-reauthorise-disclosure* [R₁₈: r-not-reauthorised($A_1, A_2, \mathcal{I}, \mathcal{G}, \{\text{T}\}$)]: A_1 requires A_2 not to transmit to other actors any document that includes information in \mathcal{I} , or authorise others to do so. For instance, Municipality expresses such requirement in the authorisation over information fiscal code and tax contributions granted to PAT.
- $TrAuth = \text{false} \rightarrow Not-reauthorised$ [R₁₈: r-not-reauthorised($A_1, A_2, \mathcal{I}, \mathcal{G}, \{\text{R, M, P, T}\}$)]: A_1 requires A_2 not to further transfer any rights when transferability of the authorisation is false (A_2 cannot transfer any permission on \mathcal{I} and for \mathcal{G}). In Figure 1, an example is the authorisation from Citizen to Municipality (dashed arrow).

Among these requirements, R₁, R₂, and R₈ can be verified only at runtime, for they require checking runtime actions that actors carry out (e.g., repudiating a delegation).

5 Detecting Conflicts at Design-Time

STS-ml models can be inconsistent. After describing the possible types of inconsistencies, we show how our reasoning framework detects conflicts among authorisations (Section 5.1), and between business policies and security requirements (Section 5.2).

5.1 Conflicts among Authorisations

We check if the stakeholders have expressed conflicting authorisations. This is non-trivial, for STS-ml models contain multiple authorisations over the same information,

and every authorisation expresses a prohibition on the operations for which rights are not transferred. The authorisation from Municipality to InfoTN allows reading (R is selected, Figure 1) information personal info, residential address and tax contributions, but prohibits the modification, production, and transmission of the given information.

Def. 1 (Authorisation conflict). Two authorisations $Auth_1 = \text{authorise}(A_1, A_2, \mathcal{I}_1, \mathcal{G}_1, \mathcal{OP}_1, TrAuth_1)$, and $Auth_2 = \text{authorise}(A_3, A_2, \mathcal{I}_2, \mathcal{G}_2, \mathcal{OP}_2, TrAuth_2)$, are conflicting (a-conflict($Auth_1, Auth_2$)) iff $\mathcal{I}_1 \cap \mathcal{I}_2 \neq \emptyset$, and either:

1. $\mathcal{G}_1 \neq \emptyset \wedge \mathcal{G}_2 = \emptyset$, or vice versa; or,
2. $\mathcal{G}_1 \cap \mathcal{G}_2 \neq \emptyset$, and either (i) $\mathcal{OP}_1 \neq \mathcal{OP}_2$, or (ii) $TrAuth_1 \neq TrAuth_2$.

□

An authorisation conflict occurs if both authorisations apply to the same information, and either (1) one authorisation restricts the permission to a goal scope, while the other does not (one implies an r-not-ntk-violated requirement, the other grants rights for any purpose); or, (2) the scopes are intersecting, and different permissions are granted (on operations, or authority to transfer). There are two authorisations to InfoTN on personal info, residential address and tax contributions: that from Municipality grants R, but prohibits M, P, and T; that from Tax Agency grants M and P, but prohibits R and T. The authorisations' scopes intersect: the goal data refined of the second authorisation is a subgoal of system maintained of the first authorisation. Thus, they are conflicting with respect to reading, modification, and production of the specified information.

5.2 Conflicts between Business Policies and Security Requirements

Given an STS-ml model without authorisation conflicts, we verify the existence of security requirements that are violated by some actor's business policy. For instance, requirement r-not-modified(Municipality, InfoTN, {personal info}) conflicts with a business policy for InfoTN that includes the relationship modify(InfoTN, goal, personal info).

Business policies define alternative strategies for an actor to fulfil her root goals. Alternatives are introduced by (i) choosing one subgoal in an or-decomposition; and (ii) deciding whether to pursue root goals that are delegated from other actors. Formally:

Def. 2 (Actor strategy). Given a business policy for an actor A , $P(A)$, an actor strategy $S_{P(A)}$ is a sub-model of $P(A)$ obtained by pruning $P(A)$ as follows:

- for every or-decomposition, only one subgoal is kept. All other subgoals are pruned, along with the elements that are reachable from the pruned subgoals only (via and/or-decompose, read/produce/modify, transmit, and delegate relationships);
- for every root goal G that is delegated to A , G can optionally be pruned.

□

In Figure 1, the business policy for TN Company Selector includes only delegated root goals. One strategy involves keeping only search module built. This goal is or-decomposed; by Def. 2, one subgoal is kept in the strategy (e.g., semantic search built). The read relationship to document high quality data is retained, as well as the document itself. An alternative strategy could, however, involve not building the search module.

We define a *variant* to enable identifying conflicts that occur only when the actors choose certain strategies. A variant combines consistently actors' strategies (each actor fulfills the root goals in her strategy), by requiring that delegated goals are in the delegatee's strategy. Also, a variant includes the authorise relationships in the model.

Def. 3 (Variant). Let M be an authorisation-consistent STS-ml model, $P(A_1), \dots, P(A_n)$ be the business policies for all actors in M . A variant of M (\mathcal{V}_M) consists of:

- a set of strategies $\{S_{P(A_1)}, \dots, S_{P(A_n)}\}$ such that, for each A_i, A_j, G , if delegate (A_i, A_j, G) is in $S_{P(A_i)}$, then G is in $S_{P(A_j)}$, and
- all the authorise relationships from M . □

Variants constrain the strategies of the actors. In Figure 1, every variant includes TN Company Selector pursuing goal search module built, for InfoTn’s root goal online system built is not delegated to it by others (thus, it has to be in her strategy), and the only possible strategy involves delegating goal search module built to TN Company Selector. Thus, there exists no variant where the latter actor does not pursue that goal.

An STS-ml model can contain more variants. TN Company Selector can choose to achieve search module built through semantic search built or enterprise search built.

Variants enable detecting conflicts between business policies and security requirements. The latter define (dis)allowed relationships for the actors’ business policies.

Def. 4 (Bus-Sec conflict). Given a variant \mathcal{V}_M , a conflict between business policies and security requirements exists if and only if, for every actor A :

- The strategy of A in \mathcal{V}_M contains one or more relationships that are prescribed by a security requirement requested from another actor A' to A ;
- The strategy of A in \mathcal{V}_M does not contain any relationships required by some requirement requested from another actor A' to A . □

Table 1 describes semi-formally how these conflicts are verified for the different types of security requirements that STS-ml supports. Below, we provide some more details.

Security Requirements. Redundancy requirements (R_3 to R_6) can be partially checked. The existence of redundant alternatives is possible, but a variant does not allow distinguishing true and fallback redundancy. Thus, true and fallback redundancy are checked the same way. Single-agent redundancy (R_3 and R_4) is fulfilled if A_2 has at least two disjoint alternatives (via or-decompositions) for G . Multi-actor redundancy (R_5 and R_6) requires that at least one alternative involves another actor A_3 . Not-redelegation (R_7) holds if there is no delegation of G or its subgoals from A_2 to other actors in the variant.

Organisational Constraints. R_9 and R_{10} require A not to or to play two roles through play relationships, respectively. R_{11} is verified if A is not the final performer for both G_1 and G_2 or their subgoals. R_{12} is verified in a similar way, but A has to be the final performer (i.e., does not delegate) for both goals.

Authorisation Requirements. These prescribe relationships that shall not be in A_2 ’s strategy in the variant. Need-to-know (R_{13}) requires no read, modify, or produce relationship on documents that make tangible some information in \mathcal{I} for some goal G' that is not in \mathcal{G} or in descendants of some goal in \mathcal{G} . R_{14} to R_{16} are verified if A_2 ’s strategy in the variant includes no read, modify, or produce relationships on documents that make tangible part of $I \in \mathcal{I}$, respectively. Non-disclosure (R_{17}) does a similar check but looking at transmissions. Non-reauthorisation (R_{18}) is fulfilled if there is no authorise relationship from A_2 to others on any operation in \mathcal{OP} over \mathcal{I} in the scope of \mathcal{G} .

Table 1. Security requirements and their design-time verification against a variant \mathcal{V}_M

Requirement	Verification at design-time
<i>Interaction requirements</i>	
$R_3 : r\text{-ts-red-ensured}(A_1, A_2, G)$	Partial. A_2 pursues goals in \mathcal{V}_M that define at least two disjoint ways to support G
$R_4 : r\text{-fs-red-ensured}(A_1, A_2, G)$	Partial. Both A_2 and another actor A_3 support G , each in a different way
$R_5 : r\text{-tm-red-ensured}(A_1, A_2, \bar{G})$	$\#\text{delegate}(A_2, A_3, \bar{G}') \in \mathcal{V}_M$. $\bar{G}' \equiv G$ or \bar{G}' is a subgoal of G
$R_6 : r\text{-fm-red-ensured}(A_1, A_2, G)$	
$R_7 : r\text{-not-redelegated}(A_1, A_2, G)$	
<i>Organisational constraints</i>	
$R_9 : r\text{-not-played-both}(STS, A, R_1, R_2)$	$\{\text{play}(A, R_1), \text{play}(A, R_2)\} \not\subseteq \mathcal{V}_M$
$R_{10} : r\text{-played-both}(STS, A, R_1, R_2)$	$\{\text{play}(A, R_1), \text{play}(A, R_2)\} \subseteq \mathcal{V}_M$
$R_{11} : r\text{-not-pursued-both}(STS, A, G_1, G_2)$	A is not the final performer for both G_1 and G_2 or their subgoals
$R_{12} : r\text{-pursued-both}(STS, A, G_1, G_2)$	A is the final performer for both G_1 and G_2 or their subgoals
<i>Authorisation requirements</i>	
$R_{13} : r\text{-not-ntk-violated}(A_1, A_2, \mathcal{I}, \mathcal{G})$	$\#\text{read}/\text{modify}/\text{produce}(A_2, G, D) \in \mathcal{V}_M$. D makes tangible (part of) $I \in \mathcal{I}$ and $G \notin \mathcal{G}$
$R_{14} : r\text{-not-read}(A_1, A_2, \mathcal{I})$	$\#\text{read}(A_2, G, D) \in \mathcal{V}_M$. D makes tangible (part of) $I \in \mathcal{I}$
$R_{15} : r\text{-not-modified}(A_1, A_2, \mathcal{I})$	$\#\text{modify}(A_2, G, D) \in \mathcal{V}_M$. D makes tangible (part of) $I \in \mathcal{I}$
$R_{16} : r\text{-not-produced}(A_1, A_2, \mathcal{I})$	$\#\text{produce}(A_2, G, D) \in \mathcal{V}_M$. D makes tangible (part of) $I \in \mathcal{I}$
$R_{17} : r\text{-not-disclosed}(A_1, A_2, \mathcal{I})$	$\#\text{transmit}(A_2, A_3, D) \in \mathcal{V}_M$. D makes tangible (part of) $I \in \mathcal{I}$
$R_{18} : r\text{-not-reauthorised}(A_1, A_2, \mathcal{I}, \mathcal{G}, \mathcal{OP})$	$\#\text{authorise}(A_2, A_3, \mathcal{I}, \mathcal{G}, \mathcal{OP}') \in \mathcal{V}_M$. $\mathcal{OP}' \subseteq \mathcal{OP}$

6 Implementation and Evaluation

STS-Tool³ [15] supports STS-ml modelling, inter-view consistency, requirements document generation, as well as automated analysis for conflict detection. Under the hood, the tool encodes STS-ml models into disjunctive Datalog programs to support our reasoning [14]. The current version of STS-Tool is the result of an iterative development process that intertwined evolutions of the language and continuous evaluations [17].

We evaluate our framework in two ways: (i) we show its effectiveness in identifying conflicts in our case study, and (ii) we conduct experiments to assess its scalability.

Findings from the Case Study. We first modelled the case study using STS-Tool (Figure 1) to then run the automated analysis to identify *authorisation conflicts*. The analysis returned several conflicts that we had not identified during the modelling, including:

- *On authority to produce*: Tax Agency authorises InfoTN to produce documents with information personal info, residential address and tax contributions to obtain refined data, whereas Municipality requires read-only access, and not production.

³ <http://www.sts-tool.eu>

- *On authority to modify:* InfoTN grants Okkam Srl the authority to modify documents with information personal info to obtain interconnected data, whereas TN Company Selector requires no document representing this information is modified.

These conflicts, which went unnoticed while modelling, originate from the stakeholders' authorisation policies. The former conflict can be resolved by negotiating the provision of adequate rights with the Municipality, while the latter can be fixed by revoking the authorisation, given that Okkam Srl does not need it (from the social view).

After fixing authorisation conflicts, we used the tool's capabilities to identify *Bus-Sec conflicts*. This activity provided us with further useful insights:

- r-not-redelegated: TN Company Selector relies on Okkam Srl to build a semantic search module (delegation of semantic search built). However, while relying on TN Company Selector, InfoTN wants this company to build the search modules, requiring it not to redelegate goal semantic search built. This interaction requirement is in conflict with the business policy on delegating semantic search built.
- r-not-modified: Engineering Tribute Srl makes an unauthorised modification of Citizen's personal info, violating the authorisation requirement r-not-modified specified by Citizen and passed on by TN Company Selector.
- r-not-produced: Citizen makes an unauthorised production of addresses, for this information is owned by the Municipality and no authorisation is granted to Citizen.
- r-not-reauthorised: Citizen wants only the Municipality to read and produce her personal info and does not allow transfer of authority, however the Municipality further authorises InfoTN to read documents with this information.
- r-pursued-both: goals semantic search built and enterprise search built should be pursued by the same actor, since a r-pursued-both normative requirement is specified between these goals. A conflict occurs because TN Company Selector is not the final performer for both goals (semantic search built is delegated to Okkam Srl).

These conflicts are due to the different policies of the companies. They can be resolved through trade-offs [4] between business policies and security requirements.

Scalability Study: Design. We have investigated how the reasoning execution time is affected by the size of the model. Taking the model in Figure 1 as a basic building block, we cloned it to obtain larger models in terms of (i) *number of elements* (nodes and relationships); and (ii) *number of variants*. The latter is motivated by our reasoning techniques, which generate STS-ml model variants (Def. 3). For details, see [14].

We ran tests on models with *zero*, *medium* and *high* variability, by customising the decomposition types in the original model. For each model, we ran the analysis 7 times, discarded the fastest and slowest executions, and computed the average execution time.

Scalability Study: Results. We have conducted our experiments on a DELL Optiplex 780 desktop PC, Pentium(R) Dual-Core CPU E5500 2.80GHz, 4Gb DDR3 399, powered by Windows 7. Below, we detail the results (summarized in Figure 2) and draw conclusions for the two scalability dimensions we have considered:

- *Number of elements* [Figure 2(a)]: we present results for all the conflict types we can detect, i.e., authorisation conflicts, violation of interaction and authorisation requirements, as well as of organisational constraints. As noticeable by the plot,

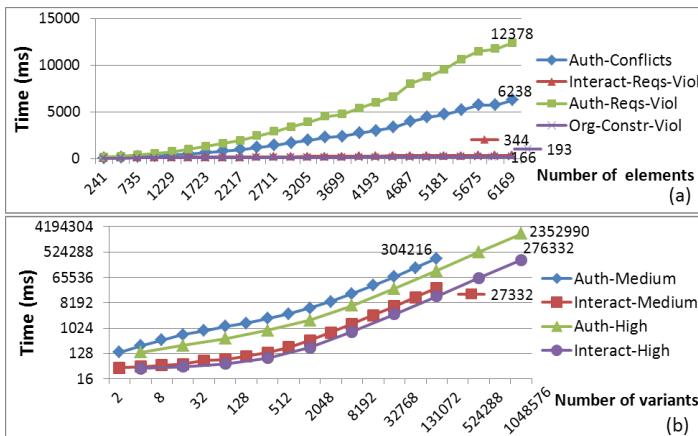


Fig. 2. Scalability results with increasing number of elements and number of variants

all techniques scale very well (linear growth). Furthermore, the tool managed to reason over our extra-large models (6,000+ elements) in about twelve seconds.

- *Number of variants* [Figure 2(b)]: this dimension affects execution time the most. We show only violations of authorisation and interaction requirements; the other checks do not increase the number of variants. While the growth is still linear in the number of variants, it is exponential in the number of elements (the model with 1,048,576 variants consists of 2,500 elements). Some medium-variability tests take longer than high-variability because, for a given number of variants, a medium-variability model contains twice the elements than a high-variability model.

The results are very promising, especially when considering that the size of real world scenarios is smaller than the extra-large models we produced with our cloning strategy.

7 Conclusions

We have proposed a framework to detect conflicts in security requirements adopting a socio-technical perspective on requirements models. Our framework is based on a revised version of STS-ml [2], the security requirements modelling language for STSs. STS-ml supports a rich set of security requirements: interaction security requirements, fine-grained authorisation requirements, and organisational constraints.

We have shown how to detect two types of conflicts: (1) among authorisation requirements; and (2) between business policies and security requirements. We have illustrated the effectiveness of our conflict identification techniques on an industrial case study, and we have reported on promising scalability results of our implementation.

Our future work includes: (1) devising further reasoning techniques to identify conflicts among security requirements (so far, we identify conflicts only among authorisation requirements); and (2) exploring possible ways to resolve the identified conflicts.

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grants no 257930 (Aniketos) and 256980 (NESSoS).

References

1. Bertino, E., Jajodia, S., Samarati, P.: A flexible authorization mechanism for relational data management systems. *ACM Transactions on Information Systems* 17(2), 101–140 (1999)
2. Dalpiaz, F., Paja, E., Giorgini, P.: Security requirements engineering via commitments. In: Proc. of STAST 2011, pp. 1–8 (2011)
3. De Landtsheer, R., van Lamsweerde, A.: Reasoning about confidentiality at requirements engineering time. In: Proc. of FSE 2005, pp. 41–49 (2005)
4. Elahi, G., Yu, E.: A goal oriented approach for modeling and analyzing security trade-offs. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 375–390. Springer, Heidelberg (2007)
5. Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J., Nuseibeh, B.: Inconsistency handling in multiperspective specifications. *IEEE TSE* 20(8), 569–578 (1994)
6. Fuxman, A., Pistore, M., Mylopoulos, J., Traverso, P.: Model checking early requirements specifications in tropos. In: Proc. of RE 2001, pp. 174–181 (2001)
7. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modeling security requirements through ownership, permission and delegation. In: Proc. of RE 2005, pp. 167–176 (2005)
8. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Reasoning with goal models. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.) ER 2002. LNCS, vol. 2503, pp. 167–181. Springer, Heidelberg (2002)
9. Haley, C.B., Laney, R., Moffett, J.D., Nuseibeh, B.: Security requirements engineering: A framework for representation and analysis. *IEEE TSE* 34(1), 133–153 (2008)
10. Horkoff, J., Yu, E.: Finding solutions in goal models: An interactive backward reasoning approach. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 59–75. Springer, Heidelberg (2010)
11. Kissel, R.: Glossary of key information security terms. Technical Report IR 7298 Rev 1, NIST (2011)
12. Liu, L., Yu, E., Mylopoulos, J.: Security and privacy requirements analysis within a social setting. In: Proc. of RE 2003, pp. 151–161 (2003)
13. Mouratidis, H., Giorgini, P.: Secure Tropos: A security-oriented extension of the tropos methodology. *IJSEKE* 17(2), 285–309 (2007)
14. Paja, E., Dalpiaz, F., Giorgini, P.: Identifying conflicts in security requirements with STS-ml. Technical Report DISI-12-041, University of Trento (2012)
15. Paja, E., Dalpiaz, F., Poggianella, M., Roberti, P., Giorgini, P.: STS-Tool: socio-technical security requirements through social commitments. In: Proc. of RE 2012, pp. 331–332 (2012)
16. Shvaiko, P., Mion, L., Dalpiaz, F., Angelini, G.: The TasLab portal for collaborative innovation. In: Proc. of ICE 2010 (2010)
17. Trösterer, S., Beck, E., Dalpiaz, F., Paja, E., Giorgini, P., Tscheligi, M.: Formative user-centered evaluation of security modeling: Results from a case study. *IJSSE* 3(1), 1–19 (2012)
18. van Lamsweerde, A., Darimont, R., Letier, E.: Managing conflicts in goal-driven requirements engineering. *IEEE TSE* 24(11), 908–926 (1998)
19. Whitman, M.E., Mattord, H.J.: Principles of Information Security, 4th edn. Course Technology Press (2011)
20. Yu, E.: Modelling strategic relationships for process reengineering. PhD thesis, University of Toronto, Canada (1996)

Optimising Conceptual Data Models through Profiling in Object Databases

Tilmann Zäschke¹, Stefania Leone², Tobias Gmünder¹, and Moira C. Norrie¹

¹ Institute for Information Systems, ETH Zurich
CH-8092 Zurich, Switzerland
{zaeschke,norrie}@inf.ethz.ch

² Semantic Information Research Laboratory, CS Department, USC
Los Angeles, CA, 90089-0781, USA
stefania.leone@usc.edu

Abstract. Agile methods promote iterative development with short cycles, where user feedback from the previous iteration is used to refactor and improve the current version. For information systems development, we propose to extend this feedback loop by using database profiling information to propose adaptations to the conceptual model to improve performance. For every software release, our database profiler identifies and analyses navigational access patterns, and proposes model optimisations based on data characteristics, access patterns and a cost-benefit model. The proposed model optimisations are based on common database and data model refactoring patterns. The database profiler has been implemented as part of an open-source object database and integrated into an existing agile development environment, where the model optimisations are presented as part of the IDE. We evaluate our approach based on an example of agile development of a research publication system.

Keywords: database profiling, object database, model optimisation.

1 Introduction

Agile methods [1] are now widely accepted in software engineering. For the development of information systems, agile practices of early adoption and frequent software releases mean that, from the first release on, depending on the project settings, users will populate the database with application-specific data. We aim to exploit the existence of such real-world data during development time to optimise and verify the conceptual data model with respect to database performance. We therefore propose an approach that complements the feedback loop inherent to agile development practices with recommendations for conceptual model optimisations based on database profiling information.

Although it is traditionally advocated that conceptual models should be independent of the underlying database materialisation, we argue that database profiling can be exploited to improve the quality of a conceptual model based on insights into how a database is used and the data accessed. In addition to

highlighting possible performance issues, it can also indicate whether the actual application usage correlates with the intended usage of the conceptual model. Comparing actual and intended usage provides opportunities for adapting and improving the model in accordance with evolving requirements and verifying whether the specified requirements match the actual usage.

We focus here on object databases (ODBMS) as the case has been made that they are well-suited to agile development [2] since they simplify the model and database evolution process. This follows from the fact that, in object databases, the conceptual model corresponds to the logical model and, unlike relational databases, there is no need for an object-relational-mapping layer.

In contrast to relational databases, ODBMS support data access through navigation as well as queries. While query optimisation is typically supported through additional data structures such as indexes, navigation is reference-based and therefore tightly bound to the conceptual model. Our approach focuses on model optimisations based on profiling navigational data access paths. For every agile software release, we track and analyse the data access paths and data characteristics in order to propose conceptual model optimisations based on model refactoring patterns and evaluated with a cost-benefit model. The recommendations can be complemented with recommendations from query and static model analysis. We implemented our profiling framework as part of the ZooDB¹ object database and integrated it into a framework for model-driven agile information system development [3]. Recommendations for model optimisations resulting from database profiling are visualised as part of the graphical model editor. As proof of concept, we developed a research publication management system with data from DBLP [4] using our agile development framework together with the extended ZooDB and measured the performance improvements resulting from the recommended optimisations.

In Sect. 2, we discuss related research and then present our approach in Sect. 3. Section 4 introduces database profiling based on navigational access paths, while the refactoring patterns and associated cost-benefit models are presented in Sect. 5. In Sect. 6 and Sect. 7, we detail the collection of profiling data and how it is brought to the developer in a model-driven development environment. A case study used to evaluate the approach is presented in Sect. 8 and concluding remarks are given in Sect. 9.

2 Background

The majority of research in database optimisation focuses on optimising the logical or physical layer of relational databases with respect to queries [5], either manually or automatically. Self-tuning databases [6] perform automated database refactorings, for example introducing a new index to support slow queries. Often, performance optimisations are discussed in the context of (de-)normalisation. While normalisation can have positive effects on database performance by avoiding redundancies and thus reducing the amount of data to be

¹ <https://github.com/tzaeschke/zoodb>

stored and loaded [7], it can introduce the need for expensive join operations which can be avoided through denormalisation [8,9]. Sanders [9] surveys denormalisation strategies and proposes guidelines and a methodology for choosing and applying denormalisation to improve database performance. In contrast to relational databases, object databases are inherently denormalised since the association construct inherent to object databases supports 1:n and n:n relationships and violates the first normal form which requires atomic attributes.

Database profiling is typically associated with the physical layer where cache hits or pages loads might be measured. Even in ODBMS, navigational data access is rarely monitored. One notable exception is Objectivity [10] which provides a profiling framework that recommends refactorings based on database profiling in the form of propositions for the selection of appropriate built-in persistent data types. Inspired by this approach, we took the idea further to generally optimise conceptual models based on database profiling information. In object databases, the models at the conceptual, logical and even physical levels are virtually equivalent which means that a single model has to fulfil requirements at all levels. Consequently, database performance can also be seen as a quality aspect of the conceptual model. Our experience with industry projects [11], as well as the case study presented in this paper, indicate that performance-related model optimisations may also result in a better understanding of the domain and ultimately an improved conceptual model. Unfortunately, we are not aware of research that underpins this observation.

Researchers have come up with different approaches for assessing the quality of conceptual models. Moody [12] presents a survey and classification of model quality approaches towards a common modelling quality standard. Following their terminology, our approach could be classified as *Observation-based (inductive)* in that database performance problems can point to, or even be seen as, modelling errors. They also state that research on conceptual model quality is dominated by static model analysis and there is little research on empirical model quality assessments such as profiling. Even the work presented in [13], which addresses performance optimisations on the conceptual model, relies purely on static model analysis and cost-benefit estimates. They propose conceptual model restructurings in the form of equivalence refactorings, which exclude replication-based refactorings, e.g. the duplication of attributes. In contrast, our approach is based on profiling the database in use and we make use of refactoring patterns to optimise performance.

Our approach was also inspired by the work presented in [14], where they formalised refactoring patterns and quality aspects to algorithmically analyse a model and propose refactorings according to given quality requirements. While they use only static model analysis, we applied a similar approach to the analysis of models based on real-world data. We made use of a number of refactoring patterns proposed in [15] and [5], identifying patterns that could be used to optimise a conceptual model with respect to navigational access, such as *Extract Class*, *Inline Class* and *Remove Middle Man*. We complemented these patterns with additional ones that proved to be effective in improving performance.

In summary, existing conceptual model quality assessments rely solely on static model analysis and database profiling is used only for optimisations on the logical or physical layers. This is no surprise considering the prevalence of relational databases. However, considering the suitability of object databases to agile methods, we take advantage of the equivalence of logical and conceptual models to use profiling for optimisation of the conceptual model. The contributions of this paper are the following. We introduce a mechanism to profile navigation paths from which we draw recommendations for conceptual model optimisations, including (navigational) refactoring recommendations. Second, we present model refactorings that can be derived by analysing database profiling data. Finally, we extend the agile development cycle with direct feedback from the database.

3 Approach

Our approach extends the agile development cycle by complementing the user feedback with recommendations for model optimisation based on database profiling. The profiling monitors navigational data access and updates, detects inefficient and performance-critical navigational access patterns and creates a set of model refactoring suggestions evaluated by a cost-benefit ratio.

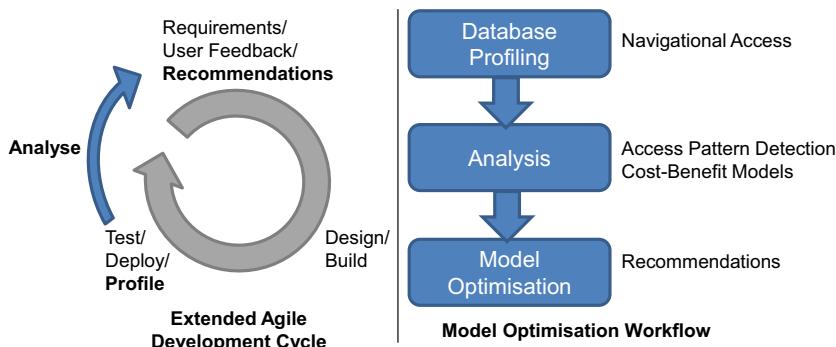


Fig. 1. Approach to Model Optimisation through Profiling

The extended agile development cycle is shown on the left of Fig. 1 and the the model optimisation workflow on the right. The cycle is extended for the test and deployment phase, where we additionally profile the database use by tracking navigational data access paths. For every accessed object, we track the accessed attributes, access mode and object predecessors in a particular navigation path. The gathered profiling information is analysed with respect to recurring data access patterns. Each detected access pattern is evaluated based on a pattern-specific cost-benefit model and, based on this evaluation, may be transformed

into recommendations for model optimisations. The recommendations are concrete model optimisations in the form of model refactorings presented to the user in a graphical and textual form.

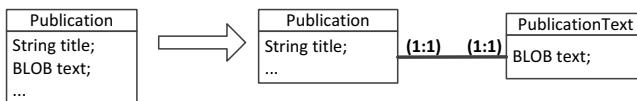


Fig. 2. Extract class: Refactoring example that extracts an attribute into a new class

Figure 2 shows an example of a model refactoring recommendation. Assume that a publication management system manages publications using the `Publication` class shown on the left. A publication defines a number of attributes, such as a title, and the actual publication is a binary large object (BLOB). Database profiling revealed that, in 90% of cases where a publication object was accessed, only the publication title was read. However, each time a publication object was loaded, the BLOB attribute was also loaded which is an expensive operation. As a consequence, our profiling framework would propose an *extract class* refactoring, illustrated in the right of Fig. 2. The refactoring suggests to split the `Publication` class into two classes, the `Publication` class and the `PublicationText` class with a single attribute `text`. The *extract class* refactoring prevents the expensive loading of the `text` attribute each time a publication object is accessed. As a result of this refactoring, the cases where data access is targeted at the publication text become slightly more expensive, since an additional navigation from the `Publication` object to the `PublicationText` object is required. Based on the frequency of these different access patterns, the expected cost and benefit of a refactoring can be evaluated. If the overall benefit of a refactoring is larger than the expected cost, the refactoring is recommended to the developer who can decide whether or not to perform the refactoring.

4 Tracking Navigational Access

For our approach, we extend database profiling so that every *navigational data access* of an application is recorded. In object databases, navigational access occurs when navigating along references between objects. These navigations form navigational access paths that can be represented as directed graphs where every node corresponds to an accessed persistent object and every edge represents a navigation along a reference. Navigational access paths typically start with a query that loads the root object of the navigation path. Once this root object is loaded, the application may use navigation to access other persistent objects. In order to obtain an application's navigational access paths, we track every access to objects and their attributes and construct a directed graph. For object access, we record frequency, average size of the objects per class and how the access occurred, for example as reference from another field in another object or via a query. For attribute access, we track the average attribute size and distinguish read and write access.

In object databases, objects are *activated*, i.e. loaded into memory, when they are accessed and, typically, the memory is cleared after each transaction. Transactions are generally rather short to prevent locking and concurrency problems. Navigational access path profiling is transactional which means that, for each transaction, such a graph structure is created and stored upon `commit()`. While the resulting graphs may be similar, recording them separately for each object allows fine-grained analysis on the object level. Note that every object can be part of several navigation graphs with different shapes, for example when it is referenced from different objects.

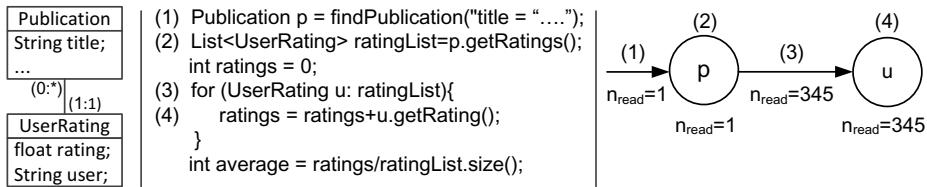


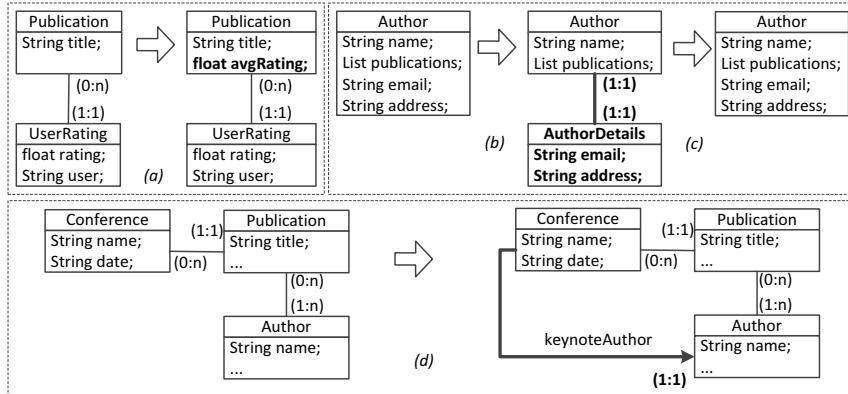
Fig. 3. Example of Navigational Access Path

An example of a UML diagram, a code snippet illustrating navigational access and the corresponding access path graph is shown in Fig. 3. In this example, publications may have user ratings. In the code snippet, the average rating of a retrieved publication is computed. In step (1), a query is performed to retrieve the publication object resulting in $n_{read} = 1$ read operations. In step (2), the user ratings are retrieved, which corresponds to an attribute access operation and $n_{read} = 1$ reads. The retrieved publication object has 345 user ratings. To calculate the average rating for the given publication, we iterate over the list of user ratings, as shown in (3). This results in $n_{read} = 345$ read operations to load the user rating objects. Next, as shown in (4), the rating attribute of each `UserRating` object is accessed, which corresponds to another $n_{read} = 345$ read operations. Finally, the average rating is computed.

5 Access Path Analysis

The goal of access path analysis is to detect recurrent, inefficient access patterns that may be improved through database refactoring. We rely on a set of well-known refactoring patterns for code [15] and databases [5] that we analysed with respect to their suitability for optimising a conceptual model with respect to database performance. We first give an overview of the supported database refactorings, before presenting how we calculate the cost-benefit ratio based on which refactorings are proposed.

Figure 4 gives an overview of a selected number of refactoring patterns which have been integrated into our profiling framework. The original model is shown on the left of each pattern and the optimised model on the right. The **Store**

**Fig. 4.** Refactoring patterns

Aggregate refactoring in Fig. 4(a) is the pattern that would have been applied to the aggregated rating example in Sect. 4. It is generally applied in cases where expensive computations of aggregates are repeatedly computed. The pattern suggests an additional attribute, in our example **Publication.avgRating**, be introduced where the computed aggregation is stored. Of course, aggregated values have to be updated and it is the responsibility of the developer to decide whether such a refactoring makes sense. The pattern **Extract Class** (b) and its inverse (c) are shown in a single example. **Extract Class** extracts one or more attributes of a class into a new class and may be based on attribute access and attribute size, as in the example presented in Sect. 3 where the publication text was extracted from the publication class since it was large and rarely accessed. Also, this pattern may be applied if the analyser identifies two or more distinct sets of attributes which are never accessed in combination. In contrast, the **Inline Class** is applied to merge two classes, e.g. if they have a 1:1 relation and attributes from both classes are typically accessed together. The **Remove Middle Man** pattern (d) recommends the insertion of a direct reference between two classes to avoid navigation over intermediate objects. In the example, a direct association was created to access the author of the leading (first) keynote for each conference, as this was a recurrent access pattern in the application.

In addition to the patterns shown in Fig. 4, we support other patterns such as the insertion of duplicate attributes, or removal of unused attributes.

In order to select and propose a refactoring pattern in the analysis phase, we calculate the cost-benefit ratio for each possible pattern. $r_{b/c} = \frac{benefit}{cost}$ is based on estimates for *cost* and *benefit* which are measured solely in terms of bytes that need to be read or written when executing a certain access pattern. This simplified approach allows us to calculate cost-benefit estimates that are almost independent of hardware, disk-layout, database size or work-load on the database. Instead, we require only a small representative dataset to suggest useful refactorings. The benefit here is that profiling can be performed by developers

with reasonable accuracy, even across software releases, without requiring access to the full databases and hardware of end-users.

We calculate the cost-benefit ratio $r_{b/c}$ for each pattern as follows:

$$r_{b/c} = \frac{benefit}{cost} = \frac{\alpha_r * \sum n_r * size_{r,avoided} + \alpha_w * \sum n_w * size_{w,avoided}}{\alpha_r * \sum n_r * size_{r,added} + \alpha_w * \sum n_w * size_{w,added}}$$

The subscripts r and w indicate read and write terms, for instance α_r and α_w are weighting factors for the different costs of reading and writing data. The benefit is the sum of all avoided operations that read or write bytes due to the refactoring. The cost is the sum of all operations that read or write additional bytes. Inside each sum, we multiply the number of occurrences n_r or n_w of a refactoring by the average number of bytes for each operation. For n and $size$, we use the access frequency and average sizes of objects and attributes that we recorded in the access graphs described in Sect. 4.

As an example, consider the *store aggregate* refactoring from Fig. 4(a) to the access path example in Fig. 3. In the refactored model, the loading of the `UserRating` objects is avoided by storing the average rating directly in the `Publication`, which corresponds to the *benefit* of the refactoring. The *cost* of the refactoring is represented by the aggregated costs that originate from the fact that the publication objects have an additional attribute, and are thus larger in size. The cost also includes cases that read `Publication` objects but do not require the `avgRating` attribute, and cases that require `avgRating` but also access the `UserRating` since they need access to the user name. Also, it includes cases that create or update `Publication` instances and cases that update `UserRating.rating`, which now have to update `Publication.avgRating` as well. For all these cases, we need to sum up the costs in terms of bytes to read or write in order to calculate the correct ratio $r_{b/c}$.

Profiling may show that a `UserRating` is 25 bytes and the `avgRating` attribute has four bytes. For simplification, we assume that the example in Fig. 3 represents the only case where publications and user ratings are accessed and that we access 15 `Publications` with 5 ratings each. Since α_r can be reduced, cost and benefit would be calculated as follows:

$$r_{b/c} = \frac{benefit}{cost} = \frac{\alpha_r * n_{r,\text{UserRating}} * size_{\text{UserRating}}}{\alpha_r * n_{r,\text{Publication}} * size_{\text{avgRating}}} = \frac{(15 * 5) * 25\text{bytes}}{15 * 4\text{bytes}} = 31.25$$

The resulting ratio $r_{c/b} = 31.25 \geq 1$ indicates that the refactoring should be suggested to the developer since the benefit outweighs the cost.

It should be noted that we do not calculate the cost-benefit ratio for combinations of refactoring patterns. If not done carefully, estimating the cost of combinations of refactoring patterns can require exponential computational complexity. Finding an efficient algorithm for simulating combinations of refactoring patterns is therefore subject to further research outside the scope of this paper.

6 Integrating Navigational Access Profiling

To realise our approach, we enhanced the open-source system ZooDB with navigational access path profiling capabilities². ZooDB is a pure Java object database that provides partial support for the JDO 3.0 standard³.

All classes to be made persistent are sub-classes of `PersistentSuperClass`. Methods that access class fields are enhanced with `activateForRead("attrName")` or `activateForWrite("attrName")`, as shown in Lst. 1.1. While currently not implemented in ZooDB, the JDO standard suggests that the code for these enhancements be injected automatically at compile time, thus relieving the developer of doing this manually.

Listing 1.1. Simplified example for automatic insertion of database callbacks.

```

1 public class Author extends PersistentSuperClass {
2     private String name;
3     public String getName() {
4         activateForRead("name");
5         return name;
6     }
7 }
```

The data structure for storing navigational access paths is shown in Fig. 5. On the left, it shows persistent application classes, such as `Author` or `Publication`, that extend `PersistentSuperClass`. Every time a persistent object is activated,

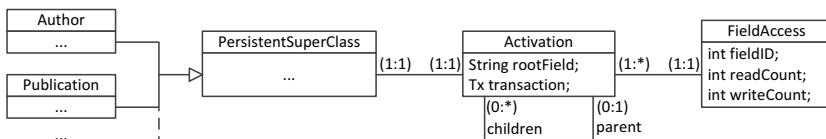


Fig. 5. UML structure for tracking object access graphs

we record the activation as an instance of `Activation`, along with the transaction id and how the activation occurred, for example as a reference from a field in another object (`rootField`) or via a query. If a second object is activated via navigation, a new instance of `Activation` is created and registered as a ‘child’ of the first `Activation` instance. Every time a field is accessed, the field access table of the corresponding `Activation` object is updated with field access details, such as number of reads and writes. If an accessed object has already been activated, the `Activation` instance is re-used and the field access counters incremented accordingly. The average sizes of objects and attributes are stored separately from these graphs in a central repository.

² The complete source code is available at <https://github.com/tzaeschke/zoodb>

³ <http://db.apache.org/jdo>

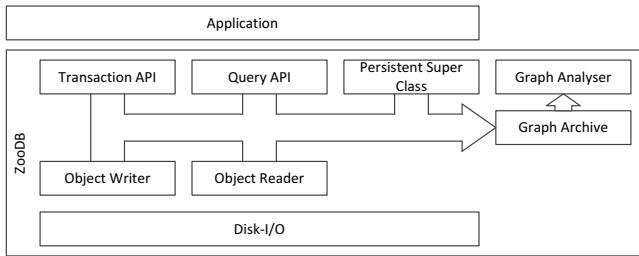


Fig. 6. ZooDB architecture overview

Figure 6 gives an overview of the architecture of ZooDB with a focus on components that were affected by the implementation of the database profiling framework. The *transaction API* and the *query API* contain callbacks for general profiling management, such as associating navigation graphs with transaction contexts or assigning queries as roots of navigation paths. The *object read* and *object write* components serialise and de-serialise objects to and from byte streams. These components were enhanced to record the number of bytes that are read or written for each object. The data collected by these components is stored in the *graph archive*, which maintains all graphs created while profiling. The *graph analyser* is executed after an application run to analyse the gathered graphs and compile recommendations for the developer. The disk-IO component, in the lower part of the figure, records the number of page reads and writes. While the number of pages is not directly used in the calculation of refactoring recommendations, it is recorded and, as shown in Sect. 8, can be used to compare the number of page-IOs before and after a refactoring to provide a hardware-independent way of estimating the usefulness of a refactoring.

After the analyser has processed the gathered profiling information, the resulting recommendations can be exported as an XML file.

7 Integration into Model-Driven Development

To validate our approach, we integrated the extended ZooDB into the AgileIS framework [3], which is an agile, model-driven development environment for information systems. AgileIS offers a graphical UML class model editor to model the persistent classes of an information system, from which application code for various target database systems, including ZooDB, can be generated [3]. The model editor provides a versioning system that allows model versions for each agile development cycle to be managed. When generating database and application code, AgileIS automatically generates schema evolution code that evolves databases between agile development cycles [16].

The new profiling capabilities in ZooDB extend the agile feedback loop with recommendations that can be integrated into application data models using AgileIS. To achieve this, AgileIS has been extended so that profiling information and recommendations from our profiling framework exported as XML can be

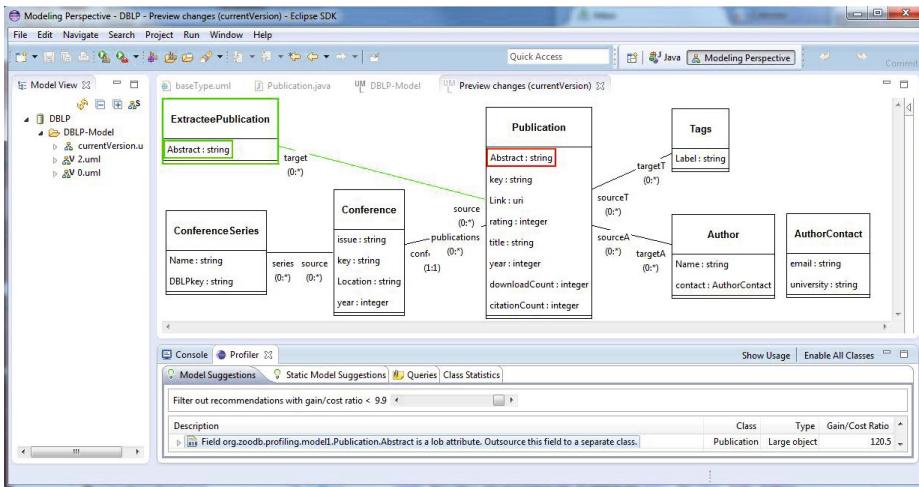


Fig. 7. The AgileIS model editor showing a suggested refactoring preview

imported into AgileIS and presented to the user through the graphical model editor. Figure 7 shows a screenshot of the recommendation preview screen of the AgileIS environment. The imported recommendations are shown textually in the lower part of the screenshot. The horizontal slider allows setting a threshold for $r_{b/c}$ below which recommendations are filtered out. In the model view in the centre, the selected recommendation is highlighted in green and red. Next to previewing recommended changes, AgileIS also supports automated application of a recommended refactoring to the current model. While such refactorings can typically be performed automatically, some refactorings require additional work by the developer. For example, for the *store aggregate* refactoring, which automatically creates an aggregate attribute, the name and type of the aggregate attribute may need to be adapted by the developer.

8 Experimental Validation

To validate our approach, we implemented an information system for managing scientific publications. Then we compared the performance of two versions of the persistent data model shown in Fig. 8. Model M_1 is the original model, for which we performed database profiling. Model M_2 is an optimised version of M_1 , achieved by applying model recommendations proposed by our profiling framework based on the profiling data of M_1 . As test data, we used a snapshot (March 2013) of the DBLP scientific publication archive [4], which we extended with abstracts that were generated as random sequences of 1000 characters. The size of the original dataset was 1.1GB consisting of 1.2 Mio. author objects, 800,000 publications etc, see Tbl. 1.

As performance tests, we ran a set of five different artificial use-cases, shown in Tbl. 2, that each consisted of a query followed by navigation operations.

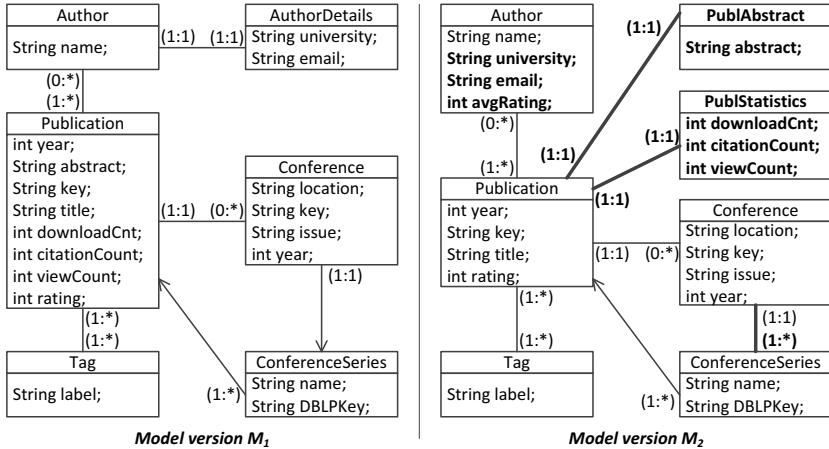


Fig. 8. Data models before (*M₁*) / after optimisation (*M₂*) (changes in bold)

The use-cases performed mostly read-only operations and were each run at least 500 times. The columns *M₁* and *M₂* in Tbl. 2 show the performance results from executing the use-cases on a database with the respective model version. When running the use-cases on *M₁*, the analyser provided a set of 28 refactoring recommendations, from which we chose 6 refactorings from the top 7 as shown in Tbl. 3, ranked by $r_{b/c}$. In Fig. 8 *M₂*, the applied refactorings are highlighted in bold. According to the recommendations, an aggregate attribute **Author.avgRating** has been added to the **Author** class and the class **AuthorDetails** has been merged with the **Author** class. Furthermore, the **Publication** class has been split into two classes **Publication** and **PublStatistics** and the **abstract** attribute was extracted into a separate class **PublAbstract**. It should be noted that we deliberately introduced a severe modelling mistake in *M₁*, where the class **ConferenceSeries** does not have a direct relation to **Conference** but only to **Publication**. This mistake affects use-case *UC₁* which retrieves conferences from a set of conference series and, as a consequence of the current model, has to iterate over the associated **Publication** objects in order to retrieve the conferences.

The results show that the suggested refactorings clearly improved database performance in terms of page reads. The biggest improvement is seen for use-case *UC₂* optimisations with a more than 100-fold improvement. The test loaded 10,000 **Author** instances from the databases and calculated for each one the average rating. For *M₁*, this involved loading all associated **Publications**, while, for *M₂*, only the authors had to be loaded. The second biggest improvement is seen for use-case *UC₁*, where the recommendations *R₁* and *R₂* from the profiling analyser fixed the modelling mistake. *R₁* and *R₂* are essentially equivalent and resulted in a single refactoring which created a (1..*) relation from **ConferenceSeries** to **Conference**. The high $r_{b/c}$ of *R₁* is a bit off since it assumed an aggregation into a single value instead of a set of **Conferences**. In this use-case, the analyser effectively detected a semantic modelling error by means of profiling a semantically correct use-case.

Table 1. Statistics of imported DBLP data

Class	Number of instances
Authors and AuthorDetails	821779
Publications (inproceedings)	1240494
Conferences	19899
ConferenceSeries	3268
Tags	55978

Table 2. Use-cases with related refactorings and average page loads in M_1 and M_2

UC_{ID}	Use-case	Related refactorings	Page loads	
			M_1	M_2
1	Get Conferences for ConferenceSeries	1, 2, 5	246577	2867
2	Get average rating of Publications per Author	4, 5	155040	1498
3	Get Publication title and year	5	35893	13535
4	Access Author and AuthorDetails together	7	17967	10341
5	Access statistics data in Publication separately	3, 5	58636	9343

Table 3. Recommended refactorings and their cost-benefit ratio $r_{b/c}$

R_{ID}	Refactoring recommendation	$r_{b/c}$
1	Aggregation Publication.conference → ConferenceSeries	88945
2	Remove Middle Man ConferenceSeries to Conference	1498
3	Split Class Publication into PublicationStatistics	1362
4	Aggregation Publication.rating → Author.avgRating	1175
5	LOB Attribute Publication.Abstract into PublAbstract	120.5
7	Merging Class AuthorDetails into Author	3.0

Other than that, while the estimated cost-benefit ratios $r_{b/c}$ did not match the test results perfectly due to interaction effects, they gave a very good indication of the usefulness of a refactoring.

9 Conclusion

We have presented an approach to optimising conceptual models based on database profiling with real-world data. With our approach, we enhance the agile development cycle with direct feedback from the database system. To realise such an approach, we have shown how an object database can be extended to track navigational access paths, from which model optimisation recommendations in the form of refactoring patterns can be derived. While we demonstrated only a limited number of refactoring patterns, our framework is extensible to support the detection and recommendation of additional patterns. Our validation has shown that recommendations with a good cost-benefit ratio resulted in a significant performance improvement when applied to the model.

While our approach may not always suggest conceptually sound refactorings, it generally yields useful information about potential mismatches between the conceptual model and the database usage, such as semantic modelling errors. We expect that, in many cases, indication of these mismatches will result in useful suggestions for further investigation that may result in changes to the model or code and thus ultimately result in an improved development process.

References

1. Fowler, M., Highsmith, J.: The Agile Manifesto. *Software Development* 9(8) (2001)
2. Ambler, S.W.: Agile Techniques for Object Databases (September 2005), <http://www.db4o.com/about/productinformation/whitepapers/>
3. Zäschke, T., Zimmerli, C., Leone, S., Nguyen, M., Norrie, M.: Adaptive Model-Driven Information Systems Development for Object Databases. In: Proc. Intl. Conf. on Information Systems Development, ISD 2011 (2011)
4. Ley, M., et al.: The DBLP Computer Science Bibliography (March 2013), <http://dblp.uni-trier.de/db/>
5. Ambler, S.W., Sadalage, P.J.: Refactoring Databases: Evolutionary Database Design. Addison-Wesley (2006)
6. Chaudhuri, S., Narasayya, V.: Self-Tuning Database Systems: A Decade Of Progress. In: Proc. 33rd Intl. Conf. on Very Large Databases, VLDB 2007 (2007)
7. Codd, E.: A Relational Model of Data for Large Shared Data Banks. *Communication of ACM* 13(6), 377–387 (1970)
8. Pinto, Y.: A Framework for Systematic Database Denormalization. *Global Journal of Computer Science and Technology* 9(4), 44–52 (2009)
9. Sanders, G., Shin, S.: Denormalization Effects on Performance of RDBMS. In: Proc. 34th Hawaii Intl. Conf. on System Sciences, HICSS 2001 (2001)
10. Objectivity: Objectivity/DB (2013), <http://www.objectivity.com/>
11. Zäschke, T., Norrie, M.C.: Revisiting Schema Evolution in Object Databases in Support of Agile Development. In: Dearle, A., Zicari, R.V. (eds.) ICOODB 2010. LNCS, vol. 6348, pp. 10–24. Springer, Heidelberg (2010)
12. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering* 55(3), 243–276 (2005)
13. Proper, H.A., Halpin, T.A.: Conceptual Schema Optimisation - Database Optimisation before sliding down the Waterfall. Technical report, Department of Computer Science, University of Queensland, Brisbane, Australia (2004)
14. Aguilera, D., Gómez, C., Olivé, A.: A method for the definition and treatment of conceptual schema quality issues. In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012. LNCS, vol. 7532, pp. 501–514. Springer, Heidelberg (2012)
15. Fowler, M., et al.: Refactoring: Improving the Design of Existing Code. Addison-Wesley (1999)
16. Zäschke, T., Leone, S., Norrie, M.C.: Optimising Schema Evolution Operation Sequences in Object Databases for Data Evolution. In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012 Main Conference 2012. LNCS, vol. 7532, pp. 369–382. Springer, Heidelberg (2012)

Skyline Queries over Incomplete Data - Error Models for Focused Crowd-Sourcing

Christoph Lofi¹, Kinda El Maarry², and Wolf-Tilo Balke²

¹ National Institute of Informatics
Tokyo 101-8430, Japan
lofi@nii.ac.jp

² Institut für Informationssysteme
Technische Universität Braunschweig, 38106 Braunschweig, Germany
{elmaarry,balke}@ifis.cs.tu-bs.de

Abstract. Skyline queries are a well-known technique for explorative retrieval, multi-objective optimization problems, and personalization tasks in databases. They are widely acclaimed for their intuitive query formulation mechanisms. However, when operating on incomplete datasets, skyline query processing is severely hampered and often has to resort to error-prone heuristics. Unfortunately, incomplete datasets are a frequent phenomenon due to widespread use of automated information extraction and aggregation. In this paper, we evaluate and compare various established heuristics for adapting skylines to incomplete datasets, focusing specifically on the error they impose on the skyline result. Building upon these results, we argue for improving the skyline result quality by employing crowd-enabled databases. This allows dynamic outsourcing of some database operators to human workers, therefore enabling the elicitation of missing values during runtime. Unfortunately, each crowd-sourcing operation will result in monetary and query runtime costs. Therefore, our main contribution is introducing a sophisticated error model, allowing us to specifically concentrate on those tuples that are highly likely to be error-prone, while relying on established heuristics for safer tuples. This technique of focused crowd-sourcing allows us to strike a perfect balance between costs and result's quality.

Keywords: Skyline Queries, Error Models, Missing Data, Crowd-Sourcing.

1 Introduction

For the last decade, skyline queries have been a popular approach for personalizing database queries. By simply providing attribute preferences, users can quickly and intuitively obtain the best items of a dataset. However, skyline queries struggle with incomplete data, a common deficiency found regularly in real world datasets. Incomplete datasets missing some values interfere with the core concept of skyline queries, the *Pareto dominance*, which basically tests two database tuples, checking whether one of them is better than or equal to the other with respect to all attribute values (if it is, the second tuple can be safely excluded from the result as it is not one of the best tuples). This test cannot be performed reliably unless all the attribute values are known.

Despite its significant real world importance, the problem of incomplete datasets received only little attention by previous research on skyline computation. Usually, heuristics are used to decide whether a tuple a dominates a tuple b when those tuples exhibit missing attribute values. These heuristics either assume some default value for the missing attribute, or slightly alter the definition of Pareto dominance. However, the focus of these heuristics is mainly on efficient computation. Their actual *quality* with respect to the “correct” Skyline set – resulting from a respective dataset without missing information – has not yet been investigated. This shortcoming will be rectified in this paper through an extensive study, relying on several real world datasets. Our results will indicate that some of the heuristics that have been commonly used for skyline computations on incomplete data induce low quality results with many false positives or false negatives, while others fare significantly better.

Building on the insights obtained in this study, we will further improve the result quality by *selectively crowd-sourcing* some tuples, i.e. completing their missing values by having human workers retrieve the real values. Here, we rely on the capabilities of *crowd-enabled databases*[1] which render this process transparent and efficient. However, every crowd-sourcing operation will incur additional costs in terms of query time and money. Therefore, we investigate a hybrid approach in which we rely on using one of the proven heuristics surveyed in our study for some tuples, and crowd-source the others. For deciding if a tuple should be handled heuristically, we introduce an *error model* reflecting those tuples that are more likely to be correctly handled by the heuristics, and those that won’t. Only those tuples for which the heuristic will likely fail are crowd-sourced, i.e. those tuples negatively impacting the correctness of the final result. By using this hybrid approach, we can find a good trade-off between result’s correctness and query costs. Therefore, the contributions in this paper can be summarized as:

- Using real-word datasets, we *present* and *evaluate* common heuristics for dealing with skyline computation on incomplete data. Here, we focus especially on skyline result correctness and quality.
- We present an *error model* for handling tuples heuristically, leading to a *hybrid approach* for efficiently combining skyline heuristics and crowd sourcing.
- We extensively *evaluate* the quality and costs of our model and show that, with just small amounts of money, the result’s correctness can be significantly improved.

2 Skyline Semantics and Skylines over Incomplete Data

In this section, we will cover the common heuristics for dealing with missing data in skyline computation, as well as the more recent ISkyline semantics [2]. Those previous works focused on computation efficiency or on reducing the size of the result set. The actual *error* induced by these heuristics compared to a real skyline computed from a complete dataset was never in the focus of attention. This issue is rectified by the survey carried out in this section, which also aims at finding a suitable baseline heuristic to be used for further improvement.

2.1 Formalizing Skyline Semantics

Skyline Queries [3] are a popular personalization technique for databases, successfully bridging set-based SQL queries and top-k style ranking queries [4]. They implement the concept of *Pareto optimality* from economics and thus allow for intuitive and simple personalization: for each relevant attribute users simply provide a preference order assuming ceteris-paribus semantics (e.g., “lower prices are preferred to higher prices given that all other attributes are equal”). Then, for any two tuples, where one tuple is preferred regarding one or more attribute(s) but equal with respect to the remaining attribute(s), rational users will always prefer the first object over the second one (the first object *Pareto dominates* the second one). Thus, the skyline set is computed by retrieving only tuples that are not dominated by any other tuple, i.e. all Pareto optimal tuples. In the basic case, skylines are computed on a complete dataset R without missing values:

Definition 1a (Complete Dataset): Formally, a dataset R is an instance of a database relation $R \subseteq D_1 \times \dots \times D_n$ on n attributes A_1, \dots, A_n with D_i as domain of attribute A_i . Each tuple t is denoted by $t := (t_1, \dots, t_m)$. For simplicity and without loss of generality, in the rest of this paper we only consider *score values* with respect to preferences, i.e. numerical domains and linearized categorical preferences normalized to $[0,1]$.

Now assume a user is stating a *skyline query*. Such a query is given by any set of preferences over the attributes of the base dataset.

Definition 2 (Numerical Preferences): A numerical preference P_i over attribute A_i with a numerical domain D_i is a *total order* over D_i . If attribute value $a \in D_i$ is preferred over value $b \in D_i$, then $(a, b) \in P_i$, also written as $a >_i b$ (“ a dominates b wrt. to P_i ”). Analogously, we define $a \gtrsim_i b$ for $a >_i b$ or $a =_i b$. Without loss of generality, we consider only *maximum score* preferences (i.e. all score values are in $[0,1]$, with 0 as worst and 1 as most preferred score).

Based on the preferences, Pareto dominance can be defined as:

Definition 3a (Pareto Dominance): We define the concept of *Pareto dominance* $t_1 >_P t_2$ between tuples $t_1, t_2 \in D_1 \times \dots \times D_n$ by t_1 dominates or is equal to t_2 with respect to *all* attributes, and t_1 dominates t_2 with respect to at least one attribute:

$$t_1 >_P t_2 \Leftrightarrow \forall i \in \{1, \dots, n\}: t_1 \gtrsim_i t_2 \wedge \exists i \in \{1, \dots, n\}: t_1 >_i t_2$$

A *skyline query* is given by a set of preferences $P = \{P_1, \dots, P_n\}$, with one preference specified for each attribute. Finally, the skyline of a dataset R is defined by:

Definition 4 (Skyline): On the complete dataset R and a set of preferences P , the skyline sky is defined as:

$$sky := skyline(R, P) = \{t_1 \in R \mid \nexists t_2 \in R : t_2 >_P t_1\}$$

Computing skyline sets is considered an expensive operation. Therefore, a variety of algorithms have been designed to significantly push the computation performance [5], e.g., by presorting [6], partitioning [7], parallelization [8], or multi-scanning the database [9–11].

2.2 Missing Data and Skylines

Missing data and incomplete datasets are becoming more and more common in modern information systems. This unfortunate development can mostly be attributed to automatically generated or aggregated datasets. The rise of Linked Open Data [12] contributes especially to this problem, where most LOD sources rely on error prone automated web scraping. Also, shopping portals and large e-commerce systems struggle hard to obtain complete datasets with all the relevant meta-data for a given product category.

Unfortunately, incomplete datasets with missing values pose a severe challenge for the original skyline semantics. When encountering a missing value during skyline computation, this basically means the test for Pareto dominance between two tuples $t_1 >_P t_2$ as given in definition 3 cannot be performed, and hence no meaningful skyline can be computed. Therefore, we will present common heuristics, which can deal with this issue in the next section. In the following, we will discuss the effects of incomplete data on skyline computation. We denote an incomplete (i.e. missing or unknown) value as \square . This leads to the following definition for incomplete datasets:

Definition 1b (Incomplete Dataset): An incomplete dataset R^\square is an instance of a database relation $R \subseteq D_1^\square \times \dots \times D_n^\square$ on n attributes A_1, \dots, A_n with D_i^\square as domain of attribute A_i using \square to denote a missing value, i.e. $D_i^\square = D_i \cup \{\square\}$. Each tuple t is denoted by $t := (t_1, \dots, t_m)$. For each tuple, at least one attribute value is known.

Similar to definition 1a, we use normalized *score values*.

The subset of all complete tuples R^C is given by $R^C := \{t \in R \mid \forall i \in \{1, \dots, m\}: t_i \neq \square\}$ and the incomplete tuples are denoted as $R^I := R^\square \setminus R^C$.

2.3 Basic Heuristics

For handling missing information in skyline computation, several commonly used basic heuristics are at hand, providing default answers for deciding a Pareto dominance test. These heuristics can be classified into two general categories, optimistic and pessimistic heuristics. Pessimistic heuristics assume that missing values actually mask inferior values, and incomplete tuples will rarely be part of the skyline. In contrast, optimistic heuristics assume that incomplete tuples might actually be very good, and thus often promote them to be part of the skyline to avoid missing out any potential good candidate. The basic heuristics covered in our study are:

- *Incompleteness as failure* (ignore incomplete tuples): This simple pessimistic heuristic just ignores all tuples with missing values. Therefore, incomplete tuples cannot dominate other tuples, nor can they be in the final result set. This heuristic is obviously quite crude, and will result in both false negatives (i.e. incomplete tuples which would have been in the skyline, if their real values were known, but are ignored by the heuristic) and false positives (i.e. complete tuples which are assumed to be in the skyline, but one of the incomplete tuples would have dominated it, if its real values were known).

- *Treat incompleteness as incomparable*: This conservative optimistic heuristic aims at minimizing false negatives, i.e. no tuple should be excluded from the skyline result unless it is clearly dominated by another tuple. As the test for dominance cannot be performed when an incomplete tuple is involved, those tuples don't dominate any other tuples, and at the same time can't be dominated. Therefore, incomplete tuples end up being in the skyline as there is no reliable information available indicating that they should be excluded. This potentially leads to many false positives, but only rarely to false negatives.
- *Surrogate with maximal values* (optimistic surrogation): This optimistic heuristic's approach differs from the two previous heuristics. Instead of providing a default decision for the dominance test, it assumes the values of missing information, i.e. it simply surrogates every missing value with the best possible value (1.0 for normalized score values). Then, the usual Pareto dominance semantics are applied for computing the skyline. The rationale behind this heuristic is that missing values are simply not known, and in the best case, they might be maximal. This heuristic may also lead to both false positives and false negatives.
- *Surrogate with minimal values* (pessimistic surrogation): This heuristic is similar to assuming maximal values, but takes a more pessimistic approach, surrogating missing values with the minimal value (e.g., 0.0). This allows incomplete tuples to be in the skyline result, but only if the tuple shows superior values for at least one of the known attributes. Therefore, this heuristic will mostly induce false negatives (incomplete tuples which should be in the skyline, but are now dominated due to the assumption of minimal values for their missing attributes).
- *Surrogate with expected values* (value imputation): This approach relies on various statistical means to predict the expected values of incomplete tuples, i.e. missing values are replaced by their estimated “real” values. The efficiency of this heuristic on skyline queries has been covered in detail in [13]. In the following, k-nearest neighbor value (KNN) imputation [14] will be used as it has been shown to be one of the stronger value prediction heuristics for general real live datasets.

2.4 ISkyline and Weak Pareto Semantics

ISkyline semantics [2] are one of the latest works centrally dealing with heuristic skyline computation over incomplete data. They closely resemble Weak Pareto Dominance published earlier in [15] and [16]. Weak Pareto and ISkyline semantics (we refer to both as simply ISkyline in the following) change the actual semantics of Pareto dominance in order to respect missing values (see def. 3b): a tuple dominates other tuples, if it is better regarding at least one attribute and at least equal or *showing missing values* in all other attributes. However, this definition implies non-transitive dominance relationships and may lead to cyclic dominance behavior. Non-transitivity poses severe problems for traditional skyline algorithms, where relying on transitivity is one of the key techniques for implementing efficient skyline computation. Accordingly, ISkyline also provides an alternative skyline computation algorithm that deals with non-transitivity.

Therefore, while using ISkyline dominance, only those in common attributes whose values for both tuples are known are considered and then traditional dominance semantics are applied. Consequently, this heuristic can also lead to larger numbers of false positives and false negatives.

Definition 3b (ISkyline Dominance): The concept of *ISkyline dominance* $t_1 >_{IS} t_2$ between two tuples $t_1, t_2 \in D_1 \times \dots \times D_n$ is given by t_1 dominates t_2 with respect to at least one attribute for which no values are missing, and for all other attributes, t_1 dominates or is equal to t_2 or one or both attribute values are missing:

$$t_1 >_{IS} t_2 \Leftrightarrow \forall i \in \{1, \dots, n\}: (t_1 = \square \vee t_2 = \square \vee t_1 \geq_i t_2) \wedge \exists i \in \{1, \dots, n\}: t_1 >_i t_2$$

2.5 Evaluation of Heuristics for Skylines on Incomplete Data

In this section, we will evaluate and compare the previously presented heuristics from a purely quality-focused point of view. This will allow us to select one heuristic that provides the highest quality results for further improvement in the second part of this paper. We basically compare the quality of a result derived from one of the heuristics with that obtained from the corresponding complete dataset.

Previous studies on skyline queries show a clear connection between skyline sizes and the degree of correlation in data [17, 18]. If the data is highly correlated, then skyline queries indeed reduce the result size drastically, fulfilling their promise of being a powerful and intuitive query personalization tool. However, if the data is anti-correlated, skyline results can easily contain 50% or even more of all database tuples. Therefore, if small skylines are to be expected, pessimistic heuristics will provide better results closer to the real results, while for anti-correlated data, optimistic approaches, which favors incomplete tuples in the skyline, will fare better.

In the upcoming evaluations, we will abstain from experimenting with synthetic data, and instead evaluate using three real-word e-commerce datasets for judging the heuristics under realistic circumstances. As with most real life datasets, our datasets also show a higher degree of correlation. All our datasets are complete, and values are artificially removed for the experiments:

a) Our first dataset is the well-known NBA player statistics (<http://www.basketballreference.com>). It consists of 21,961 tuples. For each player, we used 5 attributes, i.e. games played, points scored, rebounds, assists, and goals. We use maximum preferences (i.e. larger values are considered to be better than smaller values), resulting in a skyline of 75 tuples.

b) The second dataset contains 1,597 notebooks, crawled in 2010 from Dooyoo.de (<http://www.dooyoo.de/notebook>). This dataset features 6 attributes: CPU frequency (maximum preference), CPU type (categorical preference encoded by a score), RAM (max.), HD (max.), display size (max.), and weight (minimum preference), resulting in a skyline of 35 tuples.

c) The third dataset contains different car models, crawled from Heise.de (<http://www.heise.de/autos/neuwagenkatalog>) in 2011, including 7,755 tuples with the following attributes: price (min.), power (max.), acceleration (max.), fuel consumption (min.), CO₂ emission (min.), and taxes (min.). It results in a skyline of 268 tuples.

In order to compare and evaluate the five heuristics, values are removed randomly from the datasets, ranging from 1% (e.g. nearly-complete dataset) to 20%. So basically, we simulate incomplete datasets while retaining the complete dataset as a reference for error computation.

For measuring the actual error of a skyline set computed by a heuristic, we rely on the inverse of *Informedness* [19], a popular metric from information retrieval. Informedness quantifies how informed a computed result is when compared to a result derived by chance. The informedness measure is based on recall and inverse recall. In contrast to using recall alone, it considers error types, false positives and false negatives, while simultaneously taking into account true positives and true negatives. Therefore, it is a fair and unbiased measure.

Definition 5 (Skyline Error): Let sky_H be a skyline computed by a chosen heuristic applied to an incomplete dataset, and sky_R be the real skyline computed from the complete dataset. The error between both sets is given by (some arguments omitted):

$$error(sky_H, sky_R) = 1 - informedness(..)$$

$$informedness(sky_H, sky_R) = recall(..) + invRecall(..) - 1$$

$$recall(sky_H, sky_R) = \frac{truePositives(..)}{truePositives(..)+falseNegatives(..)}$$

$$invRecall(sky_H, sky_R) = \frac{trueNegatives(..)}{trueNegatives(..)+falsePositives(..)}$$

Focusing on the skyline error incurred by each heuristic as the missing values in the entire dataset increases from 1 to 20%, we can see a rather consistent result across the three datasets (see Figure 1). Using the ISkyline semantics [2] yields the highest skyline error, whilst using minimal value surrogation consistently results in the smallest skyline error, with the exception to the NBA dataset where surrogating with the KNN-predicted value fares better. However, this should be regarded as a special case attributed to the predictability and uniformity of that particular dataset. Also, maximal values surrogates show bad results (however, optimistic approaches are expected to be less effective for data with high correlation). The two basic heuristics that rely on default handling (ignoring incomplete tuples & incompleteness as incomparable) show similar middle ground results, even though one is optimistic and the other pessimistic. This can be attributed to their local nature which does not allow far-reaching consequences (e.g. despite being optimistic and including incomplete tuples in the skyline, the ‘incomparable’ heuristic does not allow incomplete tuples to dominate other tuples. Very much in contrast to the other optimistic maximum surrogation heuristic, which owes its bad results to complete tuples that has been wrongfully excluded.) Interestingly, the more complex KNN value imputation heuristic, which surrogates with expected values, only results in an average skyline quality.

Basically, this survey shows that from a pure quality perspective, the simple pessimistic approach surrogating all missing values with the minimal value yields the best

results, rendering all other approaches (particularly the optimistic ones) vastly inferior when focusing on the closest resemblance with the correct result.

But still, while pessimistically surrogating with minimal values does lead to better results than all the other heuristics, it severely discriminates against incomplete tuples, which consequently have only now slim chances to be included in the result. This drawback is rectified by our hybrid skyline approach, which diminishes this imbalance and further improves the result's quality by obtaining additional information using crowd-sourcing.

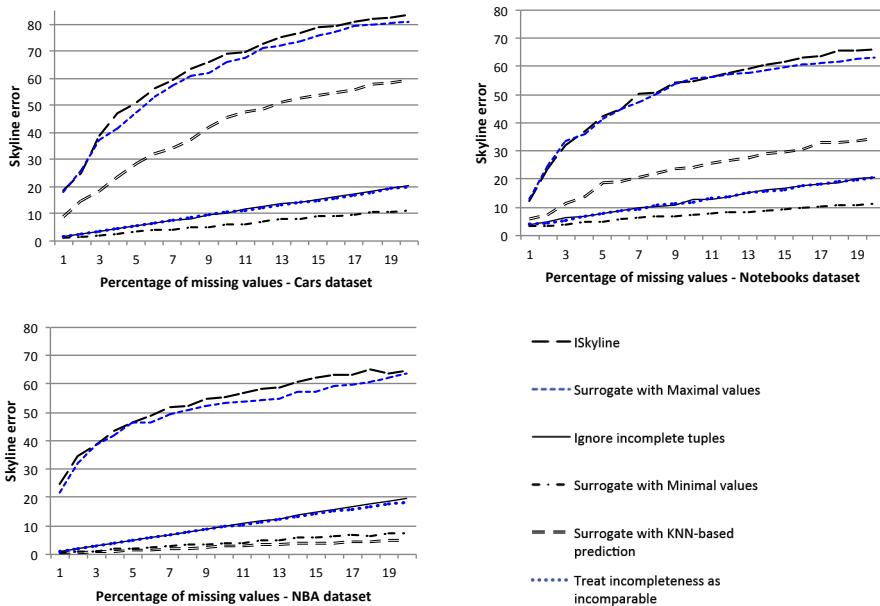


Fig. 1. Skyline error at different % of missing values for various adapting skylines to incomplete datasets heuristics

3 Improving Skyline Quality with Focused Crowd-Sourcing

In this section, using the abilities provided by Crowd-enabled DBMS, we aim at improving the skyline result's quality that was achieved by heuristic approaches for missing data through crowd-sourcing some of the missing values in a focused fashion. Crowd-enabled DBMS [1] have proven to be a very powerful and popular technology, fusing traditional relational technology with the cognitive power of people. Here, the DBMS can issue operations during runtime in the form of Human Intelligence Tasks (HITs) to crowd-sourcing services like Amazon's Mechanical Turk, CrowdFlower, or SamaSource. This technique can be used to complete the tables' missing data in a *query-driven fashion*, i.e. queries can be executed by filling empty fields despite incomplete data. Moreover, even complex cognitive tasks like reference reconciliation

can be performed, and the crowd can also be leveraged to “implement” database operators that would require cognitive abilities like scoring tuples with respect to perceived criteria (e.g., images by visual appeal), or performing perceptual comparisons (e.g., tagging images with emotions).

A major issue in crowd-sourcing is the reliability of the results obtained from the workers due to malicious or simply incompetent workers. In most cases, these concerns can be addressed effectively with quality control measures like majority voting or Gold sampling [20]. In previous studies on crowd-sourcing it has been shown that within certain bounds, missing values in database tuples can be elicited with reliable efficiency and quality as long as the information is generally available. That’s especially true for factual data that can be looked-up on the Web without requiring expert knowledge (e.g., product specifications, telephone numbers, addresses, etc.). In such a case, the expected data quality is high with only a moderate amount of quality assurance (e.g., majority votes). For example, [20] reports that crowd-sourced manual look-ups of movie genres in IMDB.com are correct in ~95% of all cases with costs of \$0.03 per tuple (including quality assurance). Accordingly, further investigations into workers’ quality are not a focus of this work.

Efficiency-wise, while each individual HIT might be cheap, costs can quickly sum up. Furthermore, each HIT requires some time for the human workers to complete the task. Because of this, the naïve approach, i.e. just crowd-sourcing all missing attribute values, is prohibitively expensive as most information that has been obtained with high costs will not even be part of the final result set.

Balancing the monetary cost and time against the desired results or improvements is therefore of utmost importance. Consequently, this section provides a hybrid approach that selectively crowd-source only the most relevant tuples, while relying on heuristics for all the others. The goal is to compute at minimal costs a skyline set that is as close as possible to the skyline which would’ve been obtained had all information been available (please note that the focus is on identifying the correct tuples and not on having a skyline result set without missing information).

3.1 Error Models for Focused Crowd-Sourcing for Skyline Queries

In order to achieve our goal of striking a favorable balance between low costs and high quality, we introduce and evaluate three ranking heuristics that rank all tuples with missing values with respect to their potential negative effects on the skyline. Next, we provide an error model for identifying only those tuples with the highest negative potential to be accordingly crowd-sourced. This enables us to tightly restrict the crowd-sourcing costs, and quickly reach at the same time a significantly better final result quality. Our resulting *two-stage approach* works as follows:

- Relying on the study results of the previous section, we surrogate all missing values with the minimal values as a baseline heuristic, i.e. all missing values are replaced by 0. This approach leads to a strong baseline in terms of quality even before crowd-sourcing. Furthermore, this heuristic has another valuable property: every tuple ending in the skyline’s result set when surrogating with 0 has a very

high probability to be a true positive (i.e. even if the real value is known, the tuple will most likely stay in the skyline). Therefore, there is no need after minimal value surrogation to crowd-source any tuple that ends up in the skyline's result.

- After the initial heuristic handling, the result's quality is improved by crowdsourcing some of the tuples to obtain their real values. Here, we can safely focus on incomplete non-skyline tuples. Furthermore, we try to crowd-source only those tuples that are most likely to be false negatives and ignore all others (i.e. ignore those which are most likely true negatives). To ultimately decide which tuples to crowd-source, the following error models aim at capturing the likeliness that an incomplete tuple is indeed a false negative.

Error Model based on Potentially Dominated Tuples: In this model, we rely on counting the number of tuples a given tuple dominates when its missing values have been surrogated. All incomplete tuples are then ranked by this count, and the top tuples (i.e. those which potentially dominate most tuples) are assumed as being most error prone and therefore crowd-sourced. Every time a tuple is crowd-sourced, this ranking is recomputed to adapt to the changes of the new information and is then reflected upon the skyline's result set. Please note that this error model, used in the second stage of our process, is independent of the heuristic used in the first stage (that chosen heuristic is only applied to tuples considered safe by the error model). We studied two variants of this error model.

a) *Min*: In the first error model, we simply count the number of dominated tuples when surrogating all missing values with the minimal value (called *minimum model* in the following). Unfortunately, this approach is less effective (see evaluations), where only few or even no other tuples are usually dominated when surrogating with minimal values. Furthermore, this variant also ignores the possible potential of tuples, as usually most real values are better than 0 (i.e. it is too pessimistic and an optimistic approach would fare better for ranking).

b) *Min-max*: Therefore, as a second variant, we temporarily (i.e. only for ranking tuples) surrogate the missing values of the current tuple with the maximum value, while retaining the minimal surrogation for all the other tuples to be ranked. This heuristic is called *min-max model* (See definition 6). This optimistic ranking heuristic leads to significantly better results due to its higher discriminating power. After crowd-sourcing, the missing values of all non-crowd-sourced tuples are again reverted to minimal values (our baseline heuristic) for the final skyline computation (i.e. optimistic handling during ranking, pessimistic handling for skyline computation of non-crowd-sourced tuples).

Definition 6 (Minimal Maximal Replacement Error Model): For a dataset $R^\square = R^C \cup R^I$ containing complete and incomplete tuples with n attributes A_1, \dots, A_n , the number of potentially dominated tuples for a given tuple $t \in R^I$ can be computed by:

$$\text{maxDomCount}(t) = |\{t_d \in R^C | \hat{t} >_P t_d\}|$$

with

$$\hat{t}_i = \begin{cases} 1 & \text{if } t_i = \square \\ t_i & \text{if } t_i \neq \square \end{cases}$$

Error model based on Impact of Missing Tuples: Not all attributes have an equal impact on the skyline result, and some attributes can be more influential for deciding a tuple's membership in the skyline than others. This supposition is quite sensible – and as will be shown in the evaluation section valid – as it mimicks the typical and common importance a user may lay on attributes when making a decision. In this error model, the potential impact of all attributes is measured in an initial dataset analysis phase. Here, we focus on measuring the skyline error introduced when a given attribute is completely ignored. Using the subset of all complete tuples, we compute the skyline. Then, we iteratively ignore each attribute, treating it as completely absent and re-compute the skyline. Comparing both skylines and computing the skyline error based on the informedness measure (see Definition 5), we get the error this attribute is responsible for introducing into the skyline's result.

This impact measure can be then combined with the previous minimal-maximal replacement heuristic to provide a better ranking, consequently improving the skyline error furthermore while still retaining similar crowd-sourcing costs (as given by def. 7). This is achieved by ranking all tuples with respect to: *Number of Dominated Tuples * (sum of the attribute impact of all missing attribute values)*. It is important to note that a tuple missing multiple attributes don't necessarily score higher than tuples with fewer missing attributes. It depends on how big the associated error of a missing attribute is, and so the sum of two missing attributes can easily be smaller than that of one highly influential attribute.

Definition 7 (Attribute Impact): For a dataset $R^\square = R^C \cup R^I$ containing complete and incomplete tuples with n attributes A_1, \dots, A_n and corresponding attribute impact error vector (I_1, \dots, I_n) , the total impact error I_t of a tuple $t \in R^I$ is given by

$$I_t := \sum_{x \in \{i | t_i = \square\}} I_x$$

Finally, all incomplete tuples $t \in R^I$ are ranked by their weighted minimal-maximal domination count:

$$\text{weightedCount}(t) = \text{maxDomCount}(t) \times I_t$$

3.2 Evaluation

Analogous to our previous evaluation in section 2.5, we now focus on evaluating how the skyline error can be further improved effectively by just few crowd-sourcing operations. First, in an initial dataset analysis phase, we measure the attributes' impact vector for each of our three datasets. Next we investigate which of the error models perform best (min value, min-max value, attribute impact with min-max value). Finally, we measure the efficiency of our approach in a real crowd-sourcing experiment.

Measuring the impact of missing attributes: In an initial dataset analysis phase on the three datasets, the following missing attribute impact's error values shown in table 1, 2 and 3 were obtained by measuring the introduced skyline's error when the corresponding attribute was ignored or completely missing. Some attributes instantly stand out, displaying a more influential role than the others. In the Notebooks dataset,

Weight, *Display Type* and *CPU Frequency* have more sway on a tuple being in the skyline than the *CPU*, which merely impacts the skyline’s quality by 8.632. Similarly in the Cars dataset the *Power* and *Price* should be carefully considered when missing. A tuple missing the value for the *Power* attribute should be treated as more highly error-prone than a tuple missing the *CO₂-Emission*.

Basic Error Model Evaluation: In this section, we evaluate our two-phase hybrid approach with both error models and crowd-sourcing.

In the first stage of our approach, we apply the heuristic surrogating with minimal values as a default handler (refer to section 2.3). For the second stage, i.e. when selecting tuples for crowd-sourcing, we focus only on tuples that could be potentially false negatives when handled heuristically. We employ our error models on these tuples to decide which of them will likely impact the result most negatively, and crowd-source these, but still handle all others heuristically. As it turns out, applying the minimal surrogation error model yields only minuscule skyline error reduction when compared to randomly crowd-sourcing tuples. Applying this model on the cars dataset with 20% missing values, the skyline error decreased from 10.8 to only 10.4 as depicted in Figure 2. This is because surrogating with minimal values is a bad heuristic for estimating the potential impact of a tuple; therefore the discriminative power of this model is rather low. Also, the model has the undesired effect that after crowd-sourcing 25 tuples, no further meaningful ranking is possible as none of the remaining incomplete tuples dominates any other tuples in the dataset.

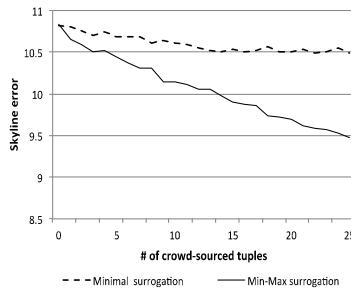


Fig. 2. Skyline improvement comparing the two counting variants

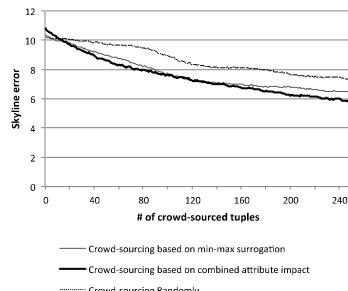


Fig. 3. Min-max counting, impact heuristic, and random crowd-sourcing

Therefore, temporarily using maximal values as surrogates as in the min-max model clearly out-performs the minimal value surrogation variant, as the skyline error decreases from 10.8 to 9.8 instead of to only 10.4 for just 25 crowd-sourcing operations. Furthermore, compared to previous studies on crowd-sourcing for skylines in [13], both these approaches significantly outperform the focused crowd-sourcing with the KNN-predicted values, where the skyline error starts high at 61.8 and decreases to only 19.8 after 25 crowd-sourcing operations. Also, when using the min-max error model, a meaningful ranking for more than 25 tuples can be created, thus allowing the focused crowd-sourcing skyline to rise to its full potential with further tangible improvements to quality.

NBA Dataset	Impact Error
Games played	62.667
Points scored	2.667
Total rebounds	50.667
Assists	78.667
Field goals made	6.667

Notebooks Dataset	Impact Error
CPU	8.632
CPU Frequency	40.181
RAM	17.323
Hard Drive	25.835
Display Type	68.571
Weight	88.571

Table 1 (left). Attribute impact error analysis for NBA dataset

Table 2 (bottom-left). Attribute impact error analysis for Notebooks dataset

Table 3 (bottom-right). Attribute impact error analysis for Cars dataset

Cars Dataset	Impact Error
Price	78.534
Power	90.312
Acceleration	33.609
Fuel Consumption	28.758
CO² Emission	10.848
Taxes	47.775

These results can be further improved by also considering the attribute impact, as illustrated in figure 3. Here, the skyline error drops from 10.8 to 5.7 upon crowd-sourcing 250 tuples out of 7,755 tuples (compared to crowd-sourcing 271 tuples when combined with min-max error models, and 358 tuples for random crowd-sourcing). Naturally, randomly crowd-sourcing just some tuples (i.e. without following any error model) leads to a very slow improvement of the result quality, and thus incurs higher costs and longer time to reach the same skyline quality achieved by either model.

Crowd-Sourcing’s Costs & Time: In this last set of experiments, we evaluate the efficiency of our approach through a real crowd-sourcing experiment, focusing on the monetary and time costs. We used CrowdFlower.com as a crowdsourcing platform, and again chose the cars dataset with 20% of missing values to run our experiment. As described in 3.1, we started with a skyline based on minimal-surrogation heuristics, and then crowd-sourced one tuple at a time relying on min-max error models with attribute impact for ranking the tuples to be crowd-sourced. And then we measured the skyline error after each crowd-sourcing operation. In order to obtain reliable values from the crowd-workers, for every crowd-sourced value, a majority vote from 4 workers was required for quality control, i.e. each single value was crowd-sourced multiple times. Thus, due to this high overhead for guaranteeing high quality results, each value cost 0.36\$ and took 1.8 minutes on average. Figure 4 and 5 illustrates the skyline result error improvement from 10.8% to 5%. In the end, crowd-sourcing 250 out of the 7,755 overall tuples roughly required 7.5 hours and 89\$.

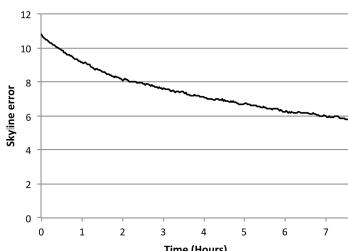


Fig. 4. Time required for CS (Cars dataset)

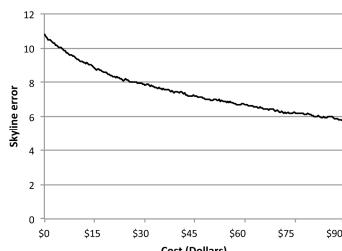


Fig. 5. Cost for CS in dollars (Cars dataset)

4 Summary and Outlook

In this paper, we extensively studied the effects of different heuristics for evaluating skyline queries on incomplete datasets with missing values. These studies used three real-life datasets, and different degrees of incompleteness have been considered. The results showed that surrogating missing values with the least desirable values shows the best results with respect to skyline correctness, while other popular approaches like treating missing values as being incomparable or the ISkyline semantics result in significantly lower skyline correctness. Building upon this insight, we developed additional error models that identify those tuples that will strongly degenerate the final result quality even when using good heuristic handling. In order to further improve result quality, we developed a hybrid approach where those tuples which severely impact quality as identified by the error model are selectively crowd-sourced to human workers in order to obtain their real values, while those tuples which are considered “safe” are handled by the respective skyline heuristic. This hybrid approach allows us to fine-tune a favorable trade-off between result quality and query costs, as just few crowd-sourced tuples can significantly improve the correctness of the skyline result.

References

1. Franklin, M., Kossmann, D., Kraska, T., Ramesh, S., Xin, R.: CrowdDB: Answering queries with crowdsourcing. In: ACM SIGMOD Int. Conf. on Management of Data, Athens, Greece (2011)
2. Khalefa, M.E., Mokbel, M.F., Levandoski, J.J.: Skyline Query Processing for Incomplete Data. In: Int. Conf. on Data Engineering (ICDE), Cancun, Mexico (2008)
3. Börzsönyi, S., Kossmann, D., Stocker, K.: The Skyline Operator. In: Int. Conf. on Data Engineering (ICDE), Heidelberg, Germany (2001)
4. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. In: Symposium on Principles of Database Systems (PODS), Santa-Barbara, California, USA (2001)
5. Godfrey, P., Shipley, R., Gryz, J.: Algorithms and analyses for maximal vector computation. The VLDB Journal 16, 5–28 (2007)
6. Bartolini, I., Ciaccia, P., Patella, M.: Efficient sort-based skyline evaluation. ACM Transactions on Database Systems 33 (2008)
7. Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive skyline computation in database systems. ACM Trans. Database Syst. 30, 41–82 (2005)
8. Selke, J., Lofi, C., Balke, W.-T.: Highly Scalable Multiprocessing Algorithms for Preference-Based Database Retrieval. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5982, pp. 246–260. Springer, Heidelberg (2010)
9. Torlone, R., Ciaccia, P.: Finding the best when it’s a matter of preference. In: 10th Italian Symposium on Advanced Database Systems (SEBD), Portoferraio, Italy (2002)
10. Boldi, P., Chierichetti, F., Vigna, S.: Pictures from Mongolia: Extracting the top elements from a partially ordered set. Theory of Computing Systems 44, 269–288 (2009)
11. Park, S., Kim, T., Park, J., Kim, J., Im, H.: Parallel skyline computation on multicore architectures. In: Int. Conf. on Data Engineering (ICDE), Shanghai, China (2009)

12. Heath, T., Hepp, M., Bizer, C.: Special Issue on Linked Data. *International Journal on Semantic Web and Information Systems (IJSWIS)* 5 (2009)
13. Lofi, C., El Maarry, K., Balke, W.-T.: Skyline Queries in Crowd-Enabled Databases. In: Int. Conf. on Extending Database Technology (EDBT), Genoa, Italy (2013)
14. Acu, E.: The treatment of missing values and its effect in the classifier accuracy. In: Classification Clustering and Data Mining Applications, pp. 1–9 (2004)
15. Balke, W.-T., Guntzer, U., Siberski, W.: Exploiting Indifference for Customization of Partial Order Skylines. In: Int. DB Engineering & Applications Symposium (IDEAS), Delhi, India (2006)
16. Balke, W.T., Guntzer, U., Siberski, W.: Restricting skyline sizes using weak Pareto dominance. *Informatik - Forschung und Entwicklung* 21, 165–178 (2007)
17. Balke, W.-T., Zheng, J.X., Guntzer, U.: Approaching the Efficient Frontier: Cooperative Database Retrieval Using High-Dimensional Skylines. In: Zhou, L.-z., Ooi, B.-C., Meng, X. (eds.) *DASFAA 2005*. LNCS, vol. 3453, pp. 410–421. Springer, Heidelberg (2005)
18. Godfrey, P.: Skyline cardinality for relational processing. In: Seipel, D., Turull-Torres, J.M. (eds.) *FoIKS 2004*. LNCS, vol. 2942, pp. 78–97. Springer, Heidelberg (2004)
19. Powers, D.M.W.: Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Flinders University Adelaide SIE07001 (2007)
20. Lofi, C., Selke, J., Balke, W.-T.: Information Extraction Meets Crowdsourcing: A Promising Couple. *Datenbank-Spektrum* 12 (2012)

Toward an Ontology-Driven Unifying Metamodel for UML Class Diagrams, EER, and ORM2

C. Maria Keet¹ and Pablo Rubén Fillottrani^{2,3}

¹ School of Mathematics, Statistics, and Computer Science,
University of KwaZulu-Natal and UKZN/CSIR-Meraka Centre for Artificial
Intelligence Research, South Africa

keet@ukzn.ac.za

² Departamento de Ciencias e Ingeniería de la Computación,
Universidad Nacional del Sur, Bahía Blanca, Argentina

prf@cs.uns.edu.ar

³ Comisión de Investigaciones Científicas, Provincia de Buenos Aires, Argentina

Abstract. Software compatibility and application integration can be achieved using their respective conceptual data models. However, each model may be represented in a different language. While such languages seem similar yet known to be distinct, no unifying framework exists that respects all of their language features. Aiming toward filling this gap, we designed a common, ontology-driven, metamodel of the static, structural, components of ER, EER, UML v2.4.1, ORM, and ORM2, such that each is a fragment of the encompassing consistent metamodel. This paper presents and overview and notable insights obtained on the real common core entities and constraints, roles and relationships, and attributes and value types that we refine with the notion of dimensional attribute.

Keywords: Metamodel, UML, EER, ORM.

1 Introduction

Complex software system design and information integration from heterogeneous sources is required due to, among others, upscaling of scientific collaboration in the life sciences [33], e-government initiatives [29], company mergers [4], and the emergence of the Semantic Web. Therefore, establishing connections between multiple conceptual models has become an important task, as the system's conceptual data models are available in, mainly, UML, EER, or ORM. However, traditional information systems development and management only exhibit this capability at the physical schema layer [7] or for conceptual models represented in the same language [2,12]. Several works has been done for conceptual data models lately, but subtle representational and expressive differences in the languages makes this task very difficult, and current tools offer only very limited functionality in linking or importing models represented in one language into one represented in another language; e.g., mandatory and disjointness is catered for, but not weak entity types, identification, or attributes [7,8,9].

The differences between the main conceptual data modelling (CDM) languages—UML Class Diagrams, ER, EER, ORM, and ORM2—may seem merely terminological, but it is known that from a metamodeling viewpoint, this is not the case [18], and at times not even within the same family of languages [25]; conversely, what may seem different may actually not be, or at least have a common ‘parent’ in meaning. The latter concerns differences in ontological foundations, but the state of the art in this area has not gone beyond a single CDM language and only for UML and ORM (e.g., [15,25]). Thus, it is unclear to what extent the languages differ or agree on their underlying principles for modelling information. This gap, in turn, is a limiting factor of mapping and transformation algorithms for CASE tools to let one work in parallel on conceptual data models represented in different languages that otherwise could be highly useful in information integration and complex system development. In addition, more detailed insight in the overlap and differences in underlying modelling principles will contribute to the understanding of the extent to which the language features affects modelling information as accurately as possible or needed, and to tools and methodologies for model development and maintenance.

To solve these issues, a comprehensive formalisation of the languages is needed to manage their interaction, but to arrive there, it first should be clear what entities and constraints exist in each language and how the differences can be reconciled without changing the languages. That is, not a comparison of metamodels, but a single integrated metamodel inclusive of all language features, so that one can unify the CDM languages and design straight-forward transformation algorithms at the conceptual layer in software and database development. We designed such a unifying metamodel for the static, structural components of UML 2.4.1 class diagrams, EER, and ORM2/FBM and their constraints, which, to the best of our knowledge, is the first of its kind. This metamodel is ontology-driven in the sense that our arguments are supported by insights from Ontology and ontologies rather than the argument of convenience to fit with an *a priori* chosen logic language. The unification brings before the differences and commonalities, such as that they all agree only on **Relationship** (association), **Role** (/association end/relationship component), and **Object type** (/class/entity type), but also ‘incomplete’ coverage of certain features in one language that are present in full in another one, such as attributes. While for the static, structural, entities ontology helped the harmonization, this is much less the case for the constraints: jointly, there are 38 constraints, but there is a remarkable small overlap among the languages (mandatory, uniqueness/functional and cardinality constraints in general, disjointness and completeness, and subset constraints). In this paper, we provide a summary of the metamodel and, due to space limitations, discuss the two most salient and ontologically interesting aspects of it, being the roles and relationships, and attributes. We discuss related works in Section 2, introduce and describe some interesting design decisions in the metamodel in Section 3, and we conclude in Section 4.

2 Related Works

There are different strands of investigation in different subfields that consider multiple CDM languages; the physical schema layer and integrating conceptual models represented in the same language have their own issues and solutions, which is beyond the scope. Comparing the languages through their metamodels (in ORM) highlighting their differences [18] is a useful step before unifying them. The Unifying Foundational Ontology (UFO) focuses on the philosophical aspects of conceptual modelling and has been applied to extend the UML 2.0 metamodel with more specific entities, such as a *Sortal Class* [15,17]. Their ontological analysis of, among others, the nature of UML's class and association did not translate into a revised metamodel specification, however.

Venable and Grundy designed [37] and implemented a partial unification in MViews [13] and Pounamu [38]. Their metamodel in the CoCoA graphical language covers a part of ER and a part of NIAM (a precursor to ORM), and omits, mainly, value types, nested entity types, and composite attributes, and NIAM is forced to have the attributes as in ER in the ‘integrated’ metamodel. Consequently, their “dynamic” ad hoc mappings are limited.

Bowers and Delcambre [7] present a framework for representing schema and data information coming from several data models, mainly relational, XML and RDF. Its main characteristic is a flat representation of schema and data, and the possibility of establishing different levels of conformance between them. However, its representational language ULD only includes ordinal, set and union class constructs, and cardinality constraints.

Boyd and McBrien [8] uses the Hypergraph Data Model to relate schemas represented in ER, relational, UML, and ORM, and includes transformation rules between them. Using graphs as intermediate representation has the advantage of providing a simple irreducible form for schemas that can be used to prove schema equivalence. The representational language includes inclusion, exclusion and union class constructs, and mandatory, unique and reflexive constraints. The combination of these types of constraints gives a rich language, but roles, aggregation, and weak entity types are missing.

Atzeni et al [2,3] describe an automatic approach that translates a schema from one model to another. They provide a small set of “metaconstructs” that can be used to characterize different models. These metaconstructs are entities (called “abstracts”), attributes (called “lexicals”), relationships, generalization, foreign keys, and complex attributes. Automatic translations between schemas are produced in the Datalog language, but translations from a rich representational language may require a sequence of such basic translations, if possible.

Thalheim [36] developed a framework for modelling layered databases, possibly integrating databases in different paradigms, such as OLAP systems and streaming databases. This type of database modelling is out of scope since we focus only on databases described with CDM languages.

Concerning unification by means of a single formalisation in a chosen logic, there are separate formalizations, which can be seen as prerequisites, and partial unifications, e.g., [1,5,9,19,22,23,27,32]. Their approach is, mainly, to choose a

logic and show it fits sufficiently with one of the CDM languages, and perhaps due to this, different logics are used for different CDM languages, therewith still not providing the sought-after interoperability for either of the languages or among each other. For instance, the Description Logic *ALUNI* is used for a partial unification [9] but *DL-Lite* and *DLR_ifa* for formalisation [1,5], which is incomplete regarding the features it supports, and ORM has features that render the language undecidable [24]. As such, they cannot simply be linked up and implemented. Once there is a comprehensive metamodel, they could be either formalised in one logic, or possibly the different logics (if implemented) could be orchestrated by means of the Distributed Ontology Language and system that is currently being standardised by ISO (<http://ontoiop.org>).

Our approach is different regarding two main aspects: scope and methodology. We aim to capture *all* the languages' constructs and *generalise* in an ontology-driven way so that the integrated metamodel subsumes the elements of EER, UML Class Diagrams v2.4.1, and ORM2 without changing the base languages. Such an integrated metamodel has as *fragments* the EER, UML Class Diagrams v2.4.1, and ORM2 metamodels, respectively, therewith leaving the base languages intact. None of the related works includes roles, aggregation, and relationship constraints, thus only limited subsets of UML or ORM are covered. Methodologically, our metamodel is ontological rather than formal, compared to all other known works that present first a formal common language for translations that leave aside important particular aspects of each language. We first develop a conceptual model of all possible entities and their relations in the selected languages, and will devise a formalization for their translations afterward. The main benefit is that it allows one to have a clear comprehension of the meaning(s) of an entity in each language whilst coping with the broader scope. This is an essential step towards achieving the full potential of information sharing.

3 Ontology-Informed Metamodelling

We focus on the *metamodel*, it being a conceptual model about the selected CDM languages such that this metamodel covers all their native features and is still consistent. Here, we do not question whether a feature of a language is a good feature or how one can make it better by using some ontological principles, but instead we aim at representing in a unified way what is present in the language. To achieve this, we use several notions from Ontology and ontologies, which serve to enhance understanding of the extant features, to reconcile or unify perceived differences, and to improve the quality of the metamodel. This does not make the metamodel an ontology, for its scope is still just the selected modelling languages.

The core entities are shown in Fig. 1 in UML Class Diagram notation and the overview of the constraints is included in Fig. 2, where a white fill of a class icon means that that entity is not present in a language, a single diagonal fill that it is present in one language, a double diagonal that it is present in two, and a dark fill that it is present in all three groups of languages (EER, UML v2.4.1, ORM2); naming conventions and terminological differences and similarities of the entities are listed in Table 1 at the end of the paper. The overview

picture of the constraints contains only those that are explicitly available in the language as graphical or textual constraint in the diagram (note that OCL is a separate OMG standard). Figures 3 and 5 are incomplete with respect to the full set of constraints that apply due to the limited expressiveness of UML Class Diagrams, but we preferred a more widely understood graphical notation for communication over a richer one, such as ORM, as it will be formalised in a suitable logic anyway. The reader may note there are a few redundancies in the metamodel; e.g., multivalued attributes can be represented by means of plain attributes. However, we aim to be complete with respect to the graphical features in the CDM languages, and for the metamodel not to judge whether it can be represented more elegantly (this can be addressed in a formalization). We describe several salient aspects of the metamodel and explain and motivate its contents in this section. More precisely, we discuss roles, relationship, and attributes (and omit from this discussion class/entity type, nested and weak entity type, subsumption, and aggregation); the metamodel for each constraint has been developed but is omitted from the discussion due to space limitations.

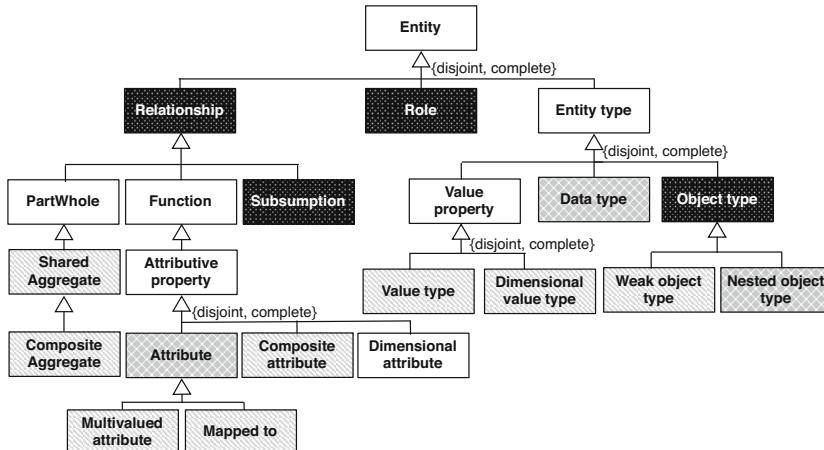


Fig. 1. Principal static entities of the metamodel; see text for details

3.1 Roles and Relationships

There are many points that can be discussed about roles and relationship, but we shall restrict ourselves to their nature and definition (not the possible types, as in [16]), and differences among them and with a object type; attributes will be discussed afterward.

A relationship, or relational property in ontology [34], is an entity that relates entities, hence, it requires at least two entities to participate in it, unlike an entity type that is a thing on its own. By this basic distinction, one can deduce that there are no unary relationships, in contrast to object types and unary predicates. The second difference between relationship and object type is due to *roles*, which are called “association ends” or “member ends” in UML [31],

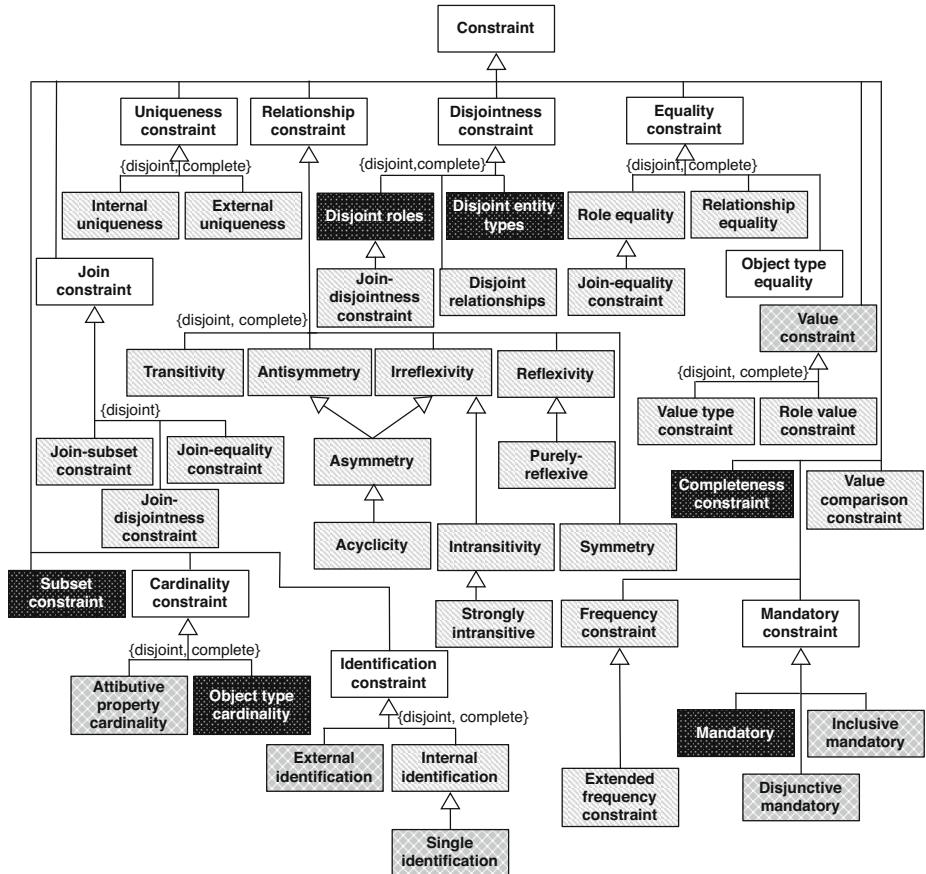


Fig. 2. Unified hierarchy of constraints in the metamodel; see text for details

“roles” in ORM and fact-based modelling [19,20,11], components of a relationship in EER [10,35]. A role is something that an object plays in a relationship, and, thus, a relationship is composed of at least two roles, therewith contributing to the characterisation of relationships and committing to the so-called *positionalist* ontological commitment of relations and relationships (see [28] for a good overview, which has been applied to ORM in [25]). The consequences are that all three CDM languages have roles—hence, logically, they have to be first-class citizens in a formalisation—and, ontologically, this inclusion entails that from an information modelling viewpoint, they form a part of the so-called ‘fundamental furniture of the universe’ and thus that they are ontologically distinct from entity types and relationships. Therefore, they appear in the metamodel as separate entities (recall Fig. 1), and Relationship, Role, and Entity Type are disjoint. The interaction between them and predicates is depicted in Fig. 3. There are three points of note. First, the relationship between Role and Relationship is a composite aggregation. Ontologically, this might be considered an

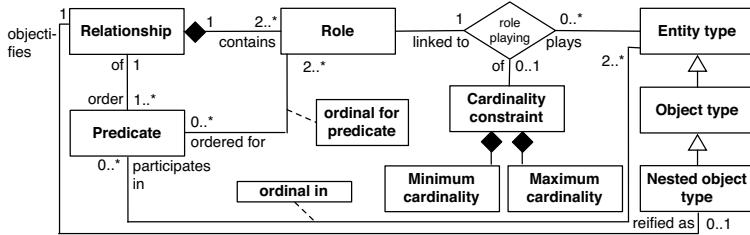


Fig. 3. Principal relationships between Relationship, Role, and Entity type; relations between roles and a predicate can only exist if there is a relation between those roles and the relationship that that predicate is an ordering of (i.e., it is a join-subset), and likewise that entities that participate in the predicate must play those roles that compose the relationship of which that predicate is an ordered version of it

underspecification, as one may argue that a relationship is even *defined* by its role-parts. Second, Fig. 3 has a ternary **role playing** between **Role**, **Entity type**, and **Cardinality constraint**: each role must have exactly one entity type with no or one cardinality constraint (where the minimum and maximum cardinality are part of the **Cardinality constraint**), and each entity type may play zero or more roles with or without declared cardinality constraint.

Third, the inclusion of predicates. While each language has roles and names for relationships, only ORM adds predicates as another way to handle its fact types. In ORM, a relationship is composed of an unordered set of roles, whereas predicates have the participating objects ordered in a fixed sequence. Given the way relationships are used in CDM languages, we restrict **Predicate** to be at least binary in the metamodel, and likewise for **Relationship**. In logic, predicates can very well be unary, but unary predicates, ontologically, are of a different type, and, besides, order in an unary predicate does not make sense. The relationship between **Relationship** and **Predicate** is not clear. The draft ORM/FBM ISO standard depicts **Predicate** with a composite aggregation to **Fact type** (**relationship**) [11]. This is incorrect, because relationship and predicate each exemplify a different ontological commitment: predicate adheres to the so-called “standard view” and the latter to “positionism” that requires the existence of roles [28], and each permutation of an ordering among participating objects without the use of roles is therewith not part of a single unordered composition of roles, and the predicates do not constitute the relationship (roles do). Yet, it is not subsumption either, because some thing without roles cannot be always a kind of something with roles. Therefore, it is included now as a plain association, where a predicate is one of the possible orderings of the entities that play the roles in that relationship. Likewise, one can argue that **Role** is not *ordered in* a, but *for* a, **Predicate**, because the roles are ordered and then ‘removed’ to obtain a predicate. Alternatively, one can look at it that roles have nothing to do with predicates, which is ontologically more precise, so that it is only **Entity type** that participates in zero or more **Predicates**. While the idea is intuitively clear and sufficient for our current purpose, ontologically, the interaction deserves refinement.

Concerning subsumption, this is straightforward for entity types, but is less clear for relationships and roles. UML 2.4.1 distinguishes between subsetting and “specialization” of associations [31]. All CDM languages support subsetting of relationships, where the participating object types are sub-ends/sub-classes of those participating in the super-relationships (or indirectly through a relationship attributes). Specialization has to do with differences in intension of the association [31], but the UML standard does not describe how that is supposed to work. The only known option to change an association’s intension, is to restrict the relational constraints [26]—e.g., each relationship that is asymmetric is also irreflexive—but little is known about its practicality, other than the few experiments in [26] for ontologies. To be comprehensive, both ways of relationship subsumption are captured in the metamodel with the more general **Subsumption**, and both UML and ORM include subsumption of roles, therefore, the participating entities for **Subsumption** are **Entity** (not shown).

3.2 Attributes, Value Types, and Data Types

From a formal perspective, it is clear what an attribute is, but this does not hold for CDM languages and ontologists have various theoretical frameworks to deal with them (albeit only partially), as discussed in [6,21]. We shall address the definition of an attribute, what Ontology and ontologies have to say about it, how it is incorporated in UML, EER, and ORM, the issue of the dimension, and how we represent it in the common metamodel and why.

Formally, an *attribute* (A) is a binary relationship between a *relationship* or *object type* ($R \cup E$) and a *data type* (D), i.e., $A \mapsto R \cup E \times D$. For instance, one can have an attribute `hasColour`, that relates an object type to a string; e.g., $\text{hasColour} \mapsto \text{Flower} \times \text{String}$. An attribute is no more, and no less.

‘Attributes’ in Ontology and Ontologies. Observe the disjointness between Object type, Value type, and Attributive property in the metamodel (Fig. 1), which reflects their meaning in the CDM languages and it also can be motivated by Ontology and ontologies. Ontology distinguishes between various kinds of properties [34] (provided one accepts the existence of properties, which the CDM languages do). Object type is in Ontology called, among others, a *sortal* property, which are, roughly, those things by means we distinguish, classify, and identify objects and they can exist on their own; e.g., `Apple`, `Person`. Attributions, or qualities, such as the `Colour` of an apple or its `Shape`, need a bearer to exist and are also formalised as unary predicates, not as attribute as in the definition above, and one cannot sort the objects by its attribution and know what those objects are (other than, e.g., ‘red objects’ or ‘square objects’). Thus, attributions differ from sortals, and philosophers agree on this matter. Philosophers do discuss about the details of the attributions, however, such as whether they are universals or tropes or a combination thereof [6,14], which is relevant for linking the metamodel to a foundational ontology, but we need not commit to either one of those for the metamodel. The distinction between object types and attributive properties is also reflected in the foundational ontologies; e.g., DOLCE has a distinction

between Quality (the attribution) and Endurant/Perdurant (the object or relation) [6] and the GFO includes this ontological distinction (albeit different from DOLCE) and has refinements for dimensional and other attributes and between atomic and non-atomic attributes [21], where the latter is, in spirit, the same as EER's composite attribute.

Attributions in UML, EER, ORM. Examples of the notations of attributions in the CDM languages are depicted in Fig. 4. UML class diagrams use the aforementioned definition and meaning of attribute and it is typically represented ‘inside’ a class- or association-icon as, e.g., “`hasColour:String`”, although `hasColour` may be drawn also as an association with at the far end a class-icon for the data type [31]. ER and EER support an *incomplete* attribute specification that involves only the ER/EER entity type and the attribute name, noting that declaring datatypes is carried out at the physical design stage of the database development. This might give the impression that an attribute is an (unary, not binary) entity of itself, but, overall, the understanding from the formal foundation of ER and EER [10,35], is that the attribute is alike UML's attribute. ER and EER contain two additional types of attributes: composite and multivalued attributes. They have been included in the metamodel (see Fig. 1) because they are in the languages, but both formally and ontologically, there is no real addition, because both can be remodelled as basic attributes.

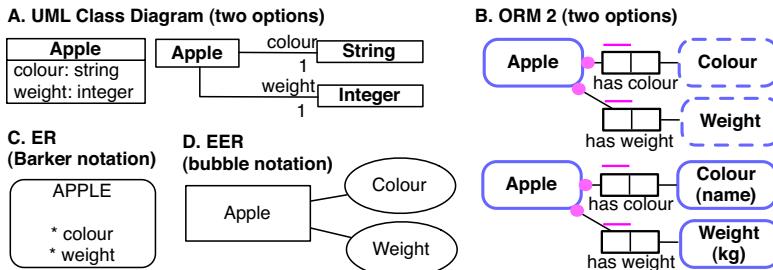


Fig. 4. Examples of attributes in UML Class Diagram notation, two ORM 2 options, and in two well-known ER/EER notations

ORM is a so-called “attribute-free” language [20], yet, actually, they do have attributes in the strict sense of the meaning. It is true that ORM does not have attributes ‘inside’ the entity type, and what is modelled as a binary in UML, like the colour of the flower, is represented in ORM as a unary *value type* `Colour` that can be related to more than one entity type through a user-defined relationship (ORM fact type), but one has to specify the data type for the value type in the ORM diagram. An ORM value type's unique feature distinguishing it from an ORM entity type (in our metamodel, Object type), is that it has a behind-the-scenes “mapped to” relationship to the datatype [11]: `mapped_to ↦ ValueType × DataType`, which is generated by the software once a value type and its data type have been declared; for our example, we obtain a

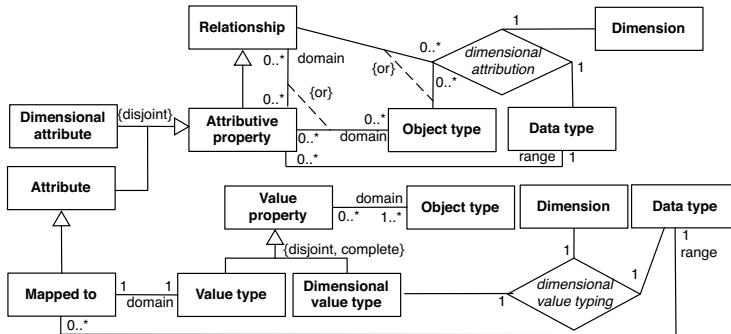


Fig. 5. Metamodel fragment for value properties and simple and attributes; Dimensional attribute is reified version of the ternary relation dimensional attribution, and likewise for Dimensional value type and dimensional value typing

mapped_to \mapsto Colour \times String in addition to a binary relationship called, say, hasColour, asserted between Flower and Colour. Thus, ORM does have attributes in the strict sense of the meaning. The principal difference between UML's and ORM's attributes, is that ORM uses three entities with two binary relationships, whereas UML collapses it all into one binary relationship.

The CDM languages' attributes can be of relationships, of object types, or both. The latter may be contentious for UML and ER/EER, as one could argue that an attribute drawn in one Object type or Relationship is different from an attribute with the same name and data type that occurs in another Object type or Relationship. Practically in UML and ER/EER tools, one indeed has to add any subsequent occurrence of an attribute again, like hasColour:String not only for apples, but also for oranges, for cars and so on. However, there is no reason why one cannot keep a separate list of attributes for the whole model, and upon reuse, select an earlier-defined one; e.g., declaring hasHeight:Integer once, and use it for both Table and Chair. This is can be done already with OWL's data properties [30] in ontology editors, and one still has the option to add another height attribute, say, hasHeight:real when one has to make that distinction for another Object type. The alternative approach is to give each attribute a unique identifier and add a constraint that it must have exactly one domain declaration that is either an object type or a relationship, i.e., an {xor} instead of an {or} in Fig. 5. Practically, one still can do this in the tools, but we keep the less constrained, more flexible way of permitting attribute reuse in the metamodel just in case in the near future such more efficient and consistent attribute management is implemented in the UML and EER CASE tools.

The Dimension. ORM's CASE tools for modelling value types, such as in NORMA or VisioModeler, lets one add not only the data type, but also, if desired, the dimension of the measurement, such as cm, kg, or day, but there is no formal characterisation of it yet. Thus, besides the basic, composite, and multivalued attributes, there may be a *dimension* for the value, i.e., there is

an implicit meaning in the values that has to do with measurements. For instance, when one needs to record the height of a plant, the measured value is not simply an integer or real, but actually denotes a value with respect to the measurement system (say, SI Units) and we measure in meters or centimeters. An attribute like `hasHeight` \mapsto `Flower` \times `Integer` does not contain any of that information, yet somehow one has to include that in the specification of the attribute or value type, even if just to facilitate data integration; e.g., as `hasHeight` \mapsto `Flower` \times `Integer` \times `cm`, or as three relations: `hasHeight` \mapsto `Flower` \times `Height`, `mapped_to` \mapsto `Height` \times `Integer`, and `hasDimension` \mapsto `Integer` \times `cm`. Within the scope of unification, we are interested in adding the notion of dimension only, not how to represent a whole system of recording measurement data for a specific scenario, because these are subject domain notions for a specific application. Therefore we chose for the more precise ternary relation `dimensional_value_typering` in our metamodel, as shown in Fig. 5, with a mandatory data type and a dimension. It has its analogue for attributes (also shown in Fig. 5), although the solution is less obvious for UML, because the standard does not mention anything about dimensions. A ternary complicates the formal apparatus, but as unification is the aim, we prefer this option, therewith essentially allowing for an extension of UML's metamodel. While the figures for ORM and UML are different due to where the attribute resides in the language, the same principle is adhered to, i.e., using optional ternaries for dimensional attributes/value types.

4 Conclusions

We presented a summary of the unifying metamodel capturing ORM, EER, and static UML v2.4.1 conceptual data modelling languages with respect to their static, structural, entities and their relationships, and their constraints. In the strict sense of the languages' features meanings, the only intersection among all these CDM languages are role, relationship (including subsumption), and object type, with each therewith also adhering to the positionalist commitment of the meaning of relationship, and for the constraints disjointness, completeness, mandatory, object cardinality, and the subset constraint. Attributions are represented differently in each language, but, ontologically, they denote the same notions. Several implicit aspects, such as dimensional attribute and its reusability and relationship versus predicate, have been made explicit.

Due to space limitations, the details of the complete metamodel is not described; it has been developed already and revealed that, aside from identification, there are no crucial irreconcilable disagreements in the languages. We have commenced with the formalisation. Once completed, this metamodel will help in the comprehension of differences between heterogenous conceptual models and in the development of tools that will aid information integration.

Acknowledgements. This work is based upon research supported by the National Research Foundation of South Africa (Project UID: 80584) and the Argentinian Ministry of Science and Technology.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyashev, M.: Reasoning over extended ER models. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 277–292. Springer, Heidelberg (2007)
2. Atzeni, P., Cappellari, P., Torlone, R., Bernstein, P.A., Gianforme, G.: Model-independent schema translation. VLDB Journal 17(6), 1347–1370 (2008)
3. Atzeni, P., Gianforme, G., Cappellari, P.: Data model descriptions and translation signatures in a multi-model framework. AMAI Mathematics and Artificial Intelligence 63, 1–29 (2012)
4. Banal-Estanol, A.: Information-sharing implications of horizontal mergers. International Journal of Industrial Organization 25(1), 31–49 (2007)
5. Berardi, D., Calvanese, D., De Giacomo, G.: Reasoning on UML class diagrams. Artificial Intelligence 168(1-2), 70–118 (2005)
6. Borgo, S., Masolo, C.: Foundational choices in DOLCE. In: Handbook on Ontologies, 2nd edn., pp. 361–381. Springer (2009)
7. Bowers, S., Delcambre, L.M.L.: Using the uni-level description (ULD) to support data-model interoperability. Data & Knowledge Engineering 59(3), 511–533 (2006)
8. Boyd, M., McBrien, P.: Comparing and transforming between data models via an intermediate hypergraph data model. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV. LNCS, vol. 3730, pp. 69–109. Springer, Heidelberg (2005)
9. Calvanese, D., Lenzerini, M., Nardi, D.: Unifying class-based representation formalisms. Journal of Artificial Intelligence Research 11, 199–240 (1999)
10. Chen, P.P.: The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems 1(1), 9–36 (1976)
11. Committee Members: Information technology – metamodel framework for interoperability (MFI) – Part xx: Metamodel for Fact Based Information Model Registration (Draft release date: 2012-04-18 2012), ISO/IEC WD 19763-xx.02
12. Fillottrani, P.R., Franconi, E., Tessaris, S.: The ICOM 3.0 intelligent conceptual modelling tool and methodology. Semantic Web Journal 3(3), 293–306 (2012)
13. Grundy, J., Venable, J.: Towards an integrated environment for method engineering. In: Proceedings of the IFIP TC8, WG8.1/8.2 Method Engineering 1996 (ME 1996), vol. 1, pp. 45–62 (1996)
14. Guizzardi, G., Masolo, C., Borgo, S.: In defense of a trope-based ontology for conceptual modeling: An example with the foundations of attributes, weak entities and datatypes. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 112–125. Springer, Heidelberg (2006)
15. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Phd thesis, University of Twente, The Netherlands. Telematica Instituut Fundamental Research Series No. 15 (2005)
16. Guizzardi, G., Wagner, G.: What's in a relationship: An ontological analysis. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 83–97. Springer, Heidelberg (2008)
17. Guizzardi, G., Wagner, G.: Using the unified foundational ontology (UFO) as a foundation for general conceptual modeling languages. In: Theory and Applications of Ontology: Computer Applications, pp. 175–196. Springer (2010)
18. Halpin, T.A.: Advanced Topics in Database Research, vol. 3, chap. Comparing Metamodels for ER, ORM and UML Data Models, pp. 23–44. Idea Publishing Group, Hershey (2004)
19. Halpin, T.: A logical analysis of information systems: static aspects of the data-oriented perspective. Ph.D. thesis, University of Queensland, Australia (1989)

20. Halpin, T.A.: Information Modeling and Relational Databases. Morgan Kaufmann Publishers, San Francisco (2001)
21. Herre, H.: General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In: Theory and Applications of Ontology: Computer Applications, ch. 14, pp. 297–345. Springer (2010)
22. ter Hofstede, A.H.M., Proper, H.A.: How to formalize it? formalization principles for information systems development methods. *Information and Software Technology* 40(10), 519–540 (1998)
23. Kaneiwa, K., Satoh, K.: Consistency checking algorithms for restricted UML class diagrams. In: Dix, J., Hegner, S.J. (eds.) FoIKS 2006. LNCS, vol. 3861, pp. 219–239. Springer, Heidelberg (2006)
24. Keet, C.M.: Prospects for and issues with mapping the Object-Role Modeling language into \mathcal{DLR}_{ifd} . In: Proc. of DL 2007. CEUR-WS, vol. 250, pp. 331–338 (2007)
25. Keet, C.M.: Positionalism of relations and its consequences for fact-oriented modelling. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 735–744. Springer, Heidelberg (2009)
26. Keet, C.M.: Detecting and revising flaws in OWL object property expressions. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d’Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 252–266. Springer, Heidelberg (2012)
27. Keet, C.M.: Ontology-driven formal conceptual data modeling for biological data analysis. In: Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data, ch. 6. Wiley (in press, 2013)
28. Leo, J.: Modeling relations. *Journal of Philosophical Logic* 37, 353–385 (2008)
29. Calo, K.M., Cenci, K.M., Fillottrani, P.R., Estevez, E.C.: Information sharing-benefits. *Journal of Computer Science & Technology* 12(2) (2012)
30. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 web ontology language structural specification and functional-style syntax. W3C Recommendation, W3C (October 27, 2009), <http://www.w3.org/TR/owl2-syntax/>
31. Object Management Group: Superstructure specification. Standard 2.4.1, Object Management Group (2012), <http://www.omg.org/spec/UML/2.4.1/>
32. Queralt, A., Teniente, E.: Decidable reasoning in UML schemas with constraints. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 281–295. Springer, Heidelberg (2008)
33. Rosenthal, A., Mork, P., Li, M.H., Stanford, J., Koester, D., Reynolds, P.: Cloud computing: a new business paradigm for biomedical information sharing. *Journal of Biomedical Informatics* 43(2), 342–353 (2010)
34. Swoyer, C.: Properties. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy, Stanford, winter 2000 edn. (2000), <http://plato.stanford.edu/archives/win2000/entries/properties/>
35. Thalheim, B.: Extended entity relationship model. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, vol. 1, pp. 1083–1091. Springer (2009)
36. Thalheim, B.: Model suites for multi-layered database modelling. In: Proceeding of the XXI Conference on Information Modelling and Knowledge Bases 2010. Frontiers in Artificial Intelligence and Applications, pp. 116–134. IOS Press (2010)
37. Venable, J., Grundy, J.: Integrating and supporting Entity Relationship and Object Role Models. In: Papazoglou, M.P. (ed.) ER 1995 and OOER 1995. LNCS, vol. 1021, pp. 318–328. Springer, Heidelberg (1995)
38. Zhu, N., Grundy, J.C., Hosking, J.G.: Pounamu: a metatool for multi-view visual language environment construction. In: IEEE Conf. on Visual Languages and Human-Centric Computing (2004)

Table 1. Terminology comparison and conventions of the entities in UML Class Diagrams, EER, and ORM/FBM (for indicative purpose)

Metamodel	UML v2.4.1	EER	ORM/FBM
Relationship	association, can be 2-ary according to the MOF 2.4.1, but also >2-ary according to the Superstructure Spec 2.4.1	relationship, ≥2-ary	atomic/compound fact type, ≥1-ary
Predicate	absent	absent	predicate
Role	association end / member end	component of a relationship	role
Entity type	classifier	absent	object type
Object type	class	entity type	non-lexical object type / entity type
Attribute	attribute	attribute, but without including a data type in the diagram	absent (represented differently)
Dimensional tribute	at-absent (no recording of dimension)	absent	absent (represented differently)
Composite attribute	more general: a property can be a composite of another property (and an attribute is a property)	composite attribute	implicitly present by adding new roles
Multivalued tribute	at-absent (represented differently)	multivalued attribute	absent (represented differently)
Value type	absent	absent	lexical object type / value type, without dimension
Dimensional value type	absent	absent	lexical object type / value type, with dimension
Data type	Data type; LiteralSpecification	absent	data type
Object subtype	subclass	subtype	subtype
Sub-relationship	subsetting or subtyping of association	subtyping the relationship (not subset constraint on fact type present in all EER variants)	
Nested object type	association class	associative entity	objectified fact type
Composite aggregate	composite aggregation	absent	absent
Shared aggregate	shared aggregation	absent	absent

Towards Ontological Foundations for the Conceptual Modeling of Events

Giancarlo Guizzardi¹, Gerd Wagner², Ricardo de Almeida Falbo¹,
Renata S.S. Guizzardi¹, and João Paulo A. Almeida¹

¹ Ontology and Conceptual Modeling Research Group (NEMO),
Federal University of Espírito Santo (UFES), Brazil

{gguizzardi,falbo,rguizzardi,jpalmeida}@inf.ufes.br

² Institute of Informatics,
Brandenburg University of Technology, Germany
G.Wagner@tu-cottbus.de

Abstract. In recent years, there has been a growing interest in the application of *foundational ontologies*, i.e., formal ontological theories in the philosophical sense, to provide a theoretically sound foundation for improving the theory and practice of conceptual modeling. In this paper, we present advances on our research on the ontological foundations of conceptual modeling by addressing the concept of events. We present a foundational ontology of events (termed *UFO-B*) together with its axiomatization in first-order logic. Moreover, we report on an implementation of *UFO-B* using the computational logic language *Alloy*, and discuss its consistency, validation and possible uses.

Keywords: Ontological Foundations for Conceptual Modeling, Formal Ontology, Ontology of Events.

1 Introduction

In recent years, there has been a growing interest in the application of formal ontological theories to provide a theoretically sound foundation for improving the theory and practice of conceptual modeling. A number of efforts have shown the benefits of ontology-based techniques in the evaluation and redesign of conceptual modeling languages [1-3], in making explicit the deep semantics of natural language [4], and, when the ontology is described formally, as a basis for validation [5] and automated reasoning [6].

The success of such ontology-based efforts depends on the availability of comprehensive ontological foundations, which have been the object of research in the past decades by several groups, under the banners of upper-level, top-level and foundational ontologies [3,7,8]. In this paper, we focus on a philosophically and cognitively well-founded reference ontology called *UFO* (*Unified Foundational Ontology*), which has been part of a long term research program on foundations for conceptual modeling [3]. *UFO* has been developed based on theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology.

The core categories of UFO have been completely formally characterized in [3]. This core fragment has been employed to analyze structural conceptual modeling constructs such as object types and taxonomic relations, associations and relations between associations, roles, properties, datatypes and weak entities, and parthood relations among objects. Moreover, it has been used to analyze, redesign and integrate reference conceptual models in a number of complex domains such as, for instance, Petroleum and Gas, Telecommunications, Software Engineering and Bioinformatics¹.

Despite a significant number of positive results in this enterprise, the focus on the core fragment of this foundational ontology has been on addressing structural conceptual modeling concepts, as opposed to dynamic ones (i.e., events and related notions). This trend can also be found in other foundational ontologies. For instance, in the BWW ontology, the treatment of the notion of events is rather minimal, i.e., an event is taken simply as a transition between states in the lawful state space of a thing [2]. However, given the importance of the notion of events for enterprise modeling [1], knowledge representation and reasoning [6], information systems engineering [9] and the semantic web [10], we argue that a widely applicable foundation for conceptual modeling requires a fuller account of the ontological notion of events, which is the subject of this paper.

The remainder of this paper is organized as follows: Section 2 presents a background on the Unified Foundational Ontology (UFO) as a context for the results developed here; Section 3 presents the main contribution of this paper, namely, a foundational ontology of events termed UFO-B with its full formal characterization; Section 4 discusses evaluation of this ontology and shows an illustrative proof of concept; Section 5 discusses related work in the literature, and, finally, Section 6 presents final considerations and a discussion on the implications of UFO-B to the practice of conceptual modeling.

2 Background: The Unified Foundational Ontology (UFO)

Like other foundational ontologies, such as DOLCE [7] and GFO [8], UFO makes a fundamental distinction between *enduring* and *perdurating* individuals (henceforth called *endurants* and *events* respectively). Classically, this distinction can be understood in terms of their behavior w.r.t. time. Endurants are said to be wholly present whenever they are present, i.e., they *are in time*, in the sense that if we say that in circumstance c_1 an endurant e has a property P_1 and in circumstance c_2 the property P_2 (possibly incompatible with P_1), it is the very same endurant e that we refer to in each of these situations. Examples of endurants are a house, a person, the Moon, an amount of sand. For instance, we can say that an individual John weights 80kg at c_1 but 68kg at c_2 . Nonetheless, we are in these two cases referring to the same individual.

Events (also called *perdurants*) are individuals composed of temporal parts. They *happen in time* in the sense that they extend in time accumulating temporal parts. Examples of events are a conversation, a football game, a symphony execution, a

¹ Related publications can be found in <http://nemo.inf.ufes.br/en/publications>

birthday party, or a particular business process. Whenever an event is present, it is not the case that all its temporal parts are present.

Among the categories of endurants, UFO makes a distinction between *objects* and *tropes* (see Fig.1). Objects are existentially independent entities. Examples include ordinary mesoscopic objects such as an individual person, a car, Alan Turing and The Rolling Stones. Tropes are endurants that are existentially dependent on other entities (termed their *bearers*) in the way in which, for example, an electrical charge can exist only in some conductor. We define a particular relation of existential dependence between tropes and their bearers called *inherence* [3]. Inherence is a type of specific constant and functional existential dependence relation. Intuitively, we have that, for instance, the headache of John can only exist if John exists and cannot be dependent on anyone but John. The notion of tropes employed here includes both what are termed *qualities* (e.g., the color of an eye, the atomic number of an atom) as well as *dispositions* [11] (e.g. the fragility of a glass, the electrical conductivity of a material).

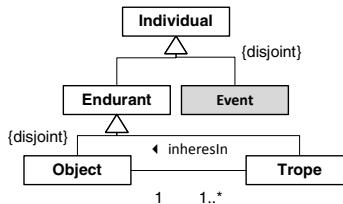


Fig. 1. A fragment of the Unified Foundational Ontology (UFO)

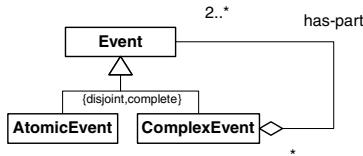
3 UFO-B: An Ontology of Events

The theory of UFO-B presented here has been fully axiomatized in standard predicate calculus and in the computational logic languages *Alloy* [12]. In the classical first-order logic axiomatization, we use a restricted quantification mechanism, which correspond to Frege's analysis of restricted quantification, i.e., $(\forall x:T A)$ is simply a shortcut for $(\forall x T(x) \rightarrow A)$ as well as $(\exists x:T A)$ is simply a shortcut for $(\exists x T(x) \wedge A)$.

In the following five subsections, we elaborate on different viewpoints of this ontology of events, namely: the mereological structure of events (Section 3.1); the participations of objects in events (Section 3.2); temporal ordering of events (Section 3.3); events as mappings from situations to situations in reality (Section 3.4); and events as manifestations of object's dispositions (Section 3.5).

3.1 Event Mereology

One first aspect of events is that events may be composed of other events. Take for instance the event e : *the murder of Caesar*. This event can be further decomposed into sub-events, namely: e_1 : the attack on Caesar, e_2 : Caesar's death. Event e_1 can, in turn, be decomposed in the events e_{11} : Caesar's restraining by the conspirators, and e_{12} : the stabbing of Caesar by Brutus. Events can be atomic or complex, depending on their mereological structure. Whilst atomic events have no proper parts, complex events are aggregations of at least two disjoint events, see Fig.2.

**Fig. 2.** Complex and atomic events

As for any mereological theory, we assume the has-part relation between events to be a strict partial order (axioms M3-M5). Moreover, we assume an atomistic mereology for events, i.e., in our theory there are events which cannot be further decomposed and, hence, are atomic w.r.t. the parthood relation [13]. We define that atomic and complex events form a (disjoint, complete) partition (axioms M1 and M2).

M1	$\forall e: \text{Event} \text{ AtomicEvent}(e) \leftrightarrow \neg \exists e': \text{Event} \text{ has-part}(e, e')$
M2	$\forall e: \text{Event} \text{ ComplexEvent}(e) \leftrightarrow \neg \text{AtomicEvent}(e)$
M3	$\forall e: \text{ComplexEvent} \neg \text{has-part}(e, e)$
M4	$\forall e, e': \text{ComplexEvent} \text{ has-part}(e, e') \rightarrow \neg \text{has-part}(e', e)$
M5	$\forall e, e': \text{ComplexEvent}, e'': \text{Event} \text{ has-part}(e, e') \wedge \text{has-part}(e', e'') \rightarrow \text{has-part}(e, e'')$

As discussed in [13], axioms M3-M5 do not suffice to characterize a parthood relation, since a number of other non-mereological relations are also partial order relations (e.g., causality, less than, temporal ordering). An additional axiom is necessary to comprise what is known as *Minimum Mereology (MM)*, namely, the so-called *Weak Supplementation Axiom (WSP)*. Intuitively, WSP states that if an object is complex then it must have at least two disjoint (i.e., non-overlapping) parts (M6). The notion of mereological overlap is defined in (M7). We here also subscribe to the view defended in [13] that an Event Mereology should commit to what is termed an *Extensional Mereology*. By including what is termed the Strong Supplementation Axiom in MM (M8), we obtain the mereological equivalent of the extensionality principle of set theory, i.e., two events are the same if they are composed of the same parts (M9).

M6	$\forall e: \text{ComplexEvent}, e': \text{Event} \text{ has-part}(e, e') \rightarrow \exists e'': \text{Event} \text{ has-part}(e, e'') \wedge \neg \text{overlaps}(e', e'')$
M7	$\forall e, e': \text{ComplexEvent} \text{ overlaps}(e, e') \leftrightarrow (\text{has-part}(e, e') \vee \text{has-part}(e', e) \vee (\exists e'': \text{Event} \text{ has-part}(e, e'') \wedge \text{has-part}(e', e'')))$
M8	$\forall e, e': \text{ComplexEvent} (\forall e'': \text{Event} \text{ has-part}(e, e'') \rightarrow \text{has-part}(e', e'')) \rightarrow ((e = e') \vee (\text{has-part}(e, e') \wedge \text{has-part}(e', e)))$
M9	$\forall e, e': \text{ComplexEvent} (e = e') \leftrightarrow (\forall e'': \text{Event} \text{ has-part}(e, e'') \leftrightarrow \text{has-part}(e', e''))$

3.2 On the Participation of Objects in Events

Events are ontologically dependent entities in the sense that they existentially depend on objects in order to exist. As previously discussed, events can be either (mereologically) atomic or complex. An atomic event is said to be *directly* existentially dependent on an object. This relation (termed here *dependsOn*) is the perdurant counterpart of the inherence relation between tropes and their bearers, i.e., *dependsOn* is a *specific constant dependence* relation [7]. Moreover, as inherence, *dependsOn* (defined for atomic events) is a functional relation (P1). A complex event is also an existentially

dependent entity. Due to the extensionality principle of the event mereology adopted here, we have that a Complex Event e' is existentially dependent on all its proper parts and, indirectly, to the objects these proper parts depend on.

The existential dependence of events on objects provides for an orthogonal way of partitioning complex events. Besides the mereological decomposition of events discussed in section 3.2, we can partition a complex event e' by separating each part of this event which is existentially dependent on each of its participants. Let us take as an example, the complex event e_{12} : *the stabbing of Caesar by Brutus*. This event can be decomposed into the events e_{Brutus} , e_{Caesar} , e_{dagger} , which depend on (in the technical sense above) Brutus, Caesar and the dagger, respectively. We here term the portion of an event which depends exclusively on a single object a *participation*. As an orthogonal way of partitioning events, participations can be atomic or complex.

In the sequel, we present an axiomatization of the notion of participation as put forth here. Firstly, we define the notion *exclusive dependence* in the following manner: an atomic event is always exclusively dependent on a single object o (P2); a complex event e' is exclusively dependent on an object o iff all its proper parts exclusively depend on o (P3).

P1	$\forall e:\text{AtomicEvent } \exists! o:\text{Object} \text{ dependsOn}(e,o)$
P2	$\forall e:\text{AtomicEvent}, o:\text{Object} \text{ excDepends}(e,o) \leftrightarrow \text{dependsOn}(e,o)$
P3	$\forall e:\text{ComplexEvent}, o:\text{Object} \text{ excDepends}(e,o) \leftrightarrow (\forall e':\text{Event hasPart}(e,e') \rightarrow \text{excDepends}(e',o))$
P4	$\forall e:\text{Event Participation}(e) \leftrightarrow \exists! o:\text{Object} \text{ excDepends}(e,o)$
P5	$\forall o:\text{Object}, p:\text{Participation participationOf}(p,o) \leftrightarrow \text{excDepends}(p,o)$

Fig. 3 summarizes the results of this subsection and depicts these two aspects on which events can be analyzed, namely, as entities with certain mereological structures, and as ontologically dependent entities consisting of a number of individual participations. As expressed in figure 3, the relations of *exclusively depends on*, *participation of* and the notion of *participation* itself are all derived notions (derived from the relations of parthood and existential dependence). Nonetheless, making explicit the notion of participation is of great importance from an ontological as well as conceptual point of view. We shall return to this point in sections 5 and 6.

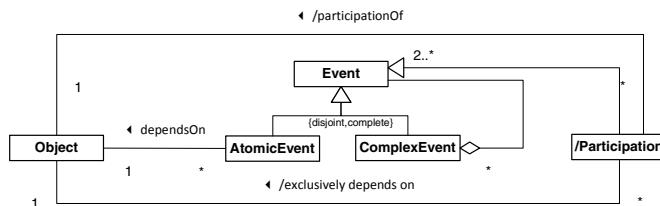


Fig. 3. Complex Events as Sums of Object's Participations

3.3 Temporal Relations between Events

As in [7], we have that all spatial properties of events are defined in terms of the spatial properties of their participants. In contrast, all temporal properties of objects are defined in terms of the events they participate in. Analogous to what has been

discussed elsewhere for objects [3], also the temporal properties of events are represented by corresponding temporal attributes taking their values from a suitable *quality structure* (or *property value space*). As discussed in depth in [3], quality structures are geometrical/topological structures associated to a property of a given type organizing the possible values that an object can assume for that property.

By decoupling a property from its value space, UFO allows for a number of alternative spaces to be associated with a given property. For instance, the property *color* of an object can be associated with a RGB cubic space as well as with the HSV color spindle [3]. Here, we take the same approach for specifying the temporal properties of events. For instance, we can have a quality structure “composed of” time intervals and time intervals themselves to be “composed of” time points. Alternatively, we can have time specified by *sui generis* entities, such as the Chronoids and Time Boundaries of GFO [8]. Moreover, we can have a model of time that admit intervals that are delimited by begin and end points as well as open intervals, continuous and non-continuous intervals, intervals with and without duration (instants). Finally, we can have models that allow for a diversity of temporal structures, such as linear, branching, parallel and circular time. Here, for illustration purposes and in order to avoid making unnecessary ontological commitments at this point, we adopt a simple but useful temporal structure: a linear order of time points. Each event is associated with two values: a begin-point and an end-point and time points are strictly ordered by a *precedes* relation. The set of temporal relations between two events corresponds to the well-known time interval relations proposed by Allen [14].

In the sequel, following [14], we present an axiomatization of these temporal Allen relations (T7-T13). In addition, we also define the linear time point space as a total order (T1-T4) and define two auxiliary functions for the begin-point and end-point of an event (T5-T6). Finally, we state that the temporal extent of an event (improperly) includes the temporal extent of all its (proper) parts (T14).

T1	$\forall t:\text{TimePoint} \neg\text{precedes}(t,t)$
T2	$\forall t,t':\text{TimePoint} \text{precedes}(t,t') \rightarrow \neg\text{precedes}(t',t)$
T3	$\forall t,t',t'':\text{TimePoint} \text{precedes}(t,t') \wedge \text{precedes}(t',t'') \rightarrow \text{precedes}(t,t'')$
T4	$\forall t,t':\text{TimePoint} (t \neq t') \rightarrow \text{precedes}(t,t') \vee \text{precedes}(t',t)$
T5	$\forall e:\text{Event} \exists !t:\text{TimePoint}, \exists !t':\text{TimePoint} (t = \text{begin-point}(e)) \wedge (t' = \text{end-point}(e))$
T6	$\forall e:\text{Event} \text{precedes}(\text{begin-point}(e), \text{end-point}(e))$
T7	$\forall e,e':\text{Event} \text{before}(e,e') \leftrightarrow \text{precedes}(\text{end-point}(e), \text{begin-point}(e'))$
T8	$\forall e,e':\text{Event} \text{meets}(e,e') \leftrightarrow (\text{end-point}(e) = \text{begin-point}(e'))$
T9	$\forall e,e':\text{Event} \text{overlaps}(e,e') \leftrightarrow \text{precedes}(\text{begin-point}(e), \text{begin-point}(e')) \wedge \text{precedes}(\text{begin-point}(e'), \text{end-point}(e)) \wedge \text{precedes}(\text{end-point}(e), \text{end-point}(e'))$
T10	$\forall e,e':\text{Event} \text{starts}(e,e') \leftrightarrow (\text{begin-point}(e) = \text{begin-point}(e')) \wedge \text{precedes}(\text{end-point}(e), \text{begin-point}(e'))$
T11	$\forall e,e':\text{Event} \text{during}(e,e') \leftrightarrow \text{precedes}(\text{begin-point}(e'), \text{begin-point}(e)) \wedge \text{precedes}(\text{end-point}(e), \text{begin-point}(e'))$
T12	$\forall e,e':\text{Event} \text{finishes}(e,e') \leftrightarrow \text{precedes}(\text{begin-point}(e'), \text{begin-point}(e)) \wedge (\text{end-point}(e) = \text{begin-point}(e'))$
T13	$\forall e,e':\text{Event} \text>equals(e,e') \leftrightarrow (\text{begin-point}(e) = \text{begin-point}(e')) \wedge (\text{end-point}(e) = \text{begin-point}(e'))$
T14	$\forall e,e':\text{Event} \text{has-part}(e,e') \rightarrow ((\text{begin-point}(e) = \text{begin-point}(e')) \vee \text{precedes}(\text{begin-point}(e), \text{begin-point}(e'))) \wedge ((\text{end-point}(e) = \text{end-point}(e')) \vee \text{precedes}(\text{end-point}(e'), \text{end-point}(e)))$

3.4 World Changes and Situations

Events are transformations from a portion of reality to another, i.e., they may change reality by changing the state of affairs from one situation to another. The notion of *situation* employed here is akin to notion of *state of affairs* in the philosophical literature. However, unlike state of affairs, situations are *bound to* specific time points. So, two qualitatively indistinguishable situations occurring at different time points are considered as numerically distinct (e.g., the situation of “John having 38°C of fever now” and “John having 38°C of fever in some moment in the past”). A situation is a particular configuration of a part of reality which can be understood as a whole. Situations can be factual or counterfactual (e.g., the situation in which “Al Gore is the president of the USA”). Factual situations are termed *Facts* [3]. Facts are situations which are said to *obtain at* particular time points.

We postulate two possible relations between situations and events: (i) a situation s *triggers* an e event, in the case that e occurs because of the obtaining of s , and; (ii) an event *brings about* a situation s , in which case the occurrence of an event e results in the situation s obtaining in the world at the time point *end-point*(e), i.e., results in s becoming a fact in *end-point*(e). A *triggers* relation between situation s and event e captures the notion that s exemplifies a state of the world that satisfies all the sufficient and necessary conditions for the manifestation of e .

A situation that *triggers* an event obtains at the begin point of that event (S1). A situation *brought about* by an event *obtains at* the end point of that event (S2). There is a unique situation that triggers a particular event occurrence (S3)². We also define that there is a unique (maximal) situation that is brought about by an event (S4), embodying the effects of the event at the moment it ends. A fact is a situation which eventually obtains (S5).

S1	$\forall s:\text{Situation}, e:\text{Event} \text{ triggers}(s,e) \rightarrow \text{obtainsIn}(s, \text{begin-point}(e))$
S2	$\forall s:\text{Situation}, e:\text{Event} \text{ brings-about}(e,s) \rightarrow \text{obtainsIn}(s, \text{end-point}(e))$
S3	$\forall e:\text{Event} \exists !s:\text{Situation} \text{ triggers}(s,e)$
S4	$\forall e:\text{Event} \exists !s:\text{Situation} \text{ brings-about}(e,s)$
S5	$\forall s:\text{Situation} \text{ fact}(s) \leftrightarrow \exists t:\text{TimePoint} \text{ obtainsIn}(s,t)$

Suppose we have a fact f that is brought about by event e . Now, suppose that f triggers event e' . In this case, we can state that the occurrence of e' is *caused by* the occurrence of e . In other words, we can state that e *directly-causes* e' iff:

S6	$\forall e,e':\text{Event} \text{ directly-causes}(e,e') \leftrightarrow \exists s:\text{Situation} \text{ brings-about}(e,s) \wedge \text{triggers}(s,e')$
----	---

Finally, we define a *causes* (S7) relation between events as follows:

S7	$\forall e,e'':\text{Event} \text{ causes}(e,e'') \leftrightarrow \text{directly-causes}(e,e'') \vee (\exists e':\text{Event} \text{ causes}(e,e') \wedge \text{causes}(e',e''))$
----	---

² A situation could be part of other (complex) situations. We refrain from discussing issues regarding the mereology of situations; this would be part of a full theory of situations which is outside the scope of this paper.

Given our characterization of temporal intervals, as well as formulae S1-S7 above, we can demonstrate that *causes* is a strict partial order. Firstly, given that situations are bound to time points, we can easily show that, given three different events e, e' and e'' , if we have *directly-causes*(e, e') and *directly-causes*(e', e''), we cannot have that *directly-causes*(e, e''). The restricted form of causation defined in S7 is also considered here to be a transitive relation. Secondly, the irreflexivity of this relation is rather straightforwardly shown from the constraint that a situation can only be bound to one time point plus the constraint that the two external time boundaries of an event are necessarily distinct and strictly ordered (as in [15], we consider here that there are no zero length events). Moreover, given this constraint, it is easy to show that the *causes* relation is asymmetric. Finally, one can show that, given distinct events e, e' and e'' , if *directly-causes*(e, e') and *directly-causes*(e', e''), then we have that e and e' *meet* (as well as e' and e'') but also that e *before* e'' .

A fragment of UFO-B summarizing the discussion in this section is depicted in Figure 4 below.

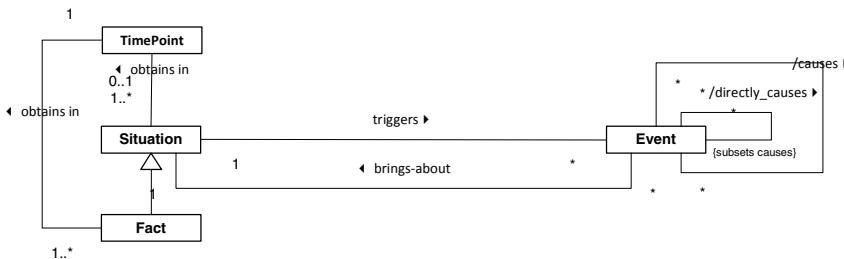


Fig. 4. Situations as parts of a world that obtain in particular time points; events as changes

3.5 Events as Manifestations of Object Dispositions

Since its initial versions, UFO's notion of particularized tropes includes both *qualities* (e.g., color, weight, temperature, electric charge) and *dispositions* (e.g., the fragility of a glass, the disposition of a magnet to attract metallic material) [3]. Following [11], we consider dispositions as properties that are only manifested in particular situations and that can also fail to be manifested. When manifested, they are manifested through the occurrence of events. Take for example the disposition of a magnet m to attract metallic material. The object m has this disposition even if it is never manifested, for example, because it is never close to any magnetic material. Nonetheless, m can certainly be said to possess that intrinsic (even essential, in this case) property, which it shares with other magnets. Now, a particular metallic material has also the disposition of being attracted by magnets. Given a situation in which m is in the presence of a particular metallic object (at a certain distance, of a certain mass, in a surface with a certain friction, etc.), the dispositions of these two entities (metallic object, magnet) can be manifested through the occurrence of a complex event, namely, the movement of that object towards the magnet.

The following constraints hold for the view of dispositions assumed here. Firstly, we specify that, as other particularized properties (tropes), dispositions are existentially dependent and therefore *inhere* in particular objects (D1). Moreover, the events we consider in this paper are manifestations of dispositions (D2). A situation triggers an event when this situation *activates* the disposition that is manifested by that event (D3). Finally, given these assumptions, we have that a particular atomic event is existentially dependent on that particular object because that event is a manifestation of a disposition of that object (which, like any intrinsic property, is entity specific) (D4).

D1	$\forall d:\text{Disposition } \exists o:\text{Object } \text{inheresIn}(d,o)$
D2	$\forall e:\text{AtomicEvent } \exists d:\text{Disposition } \text{manifestedBy}(d,e)$
D3	$\forall s:\text{Situation}, e:\text{AtomicEvent} \text{ triggers}(s,e) \leftrightarrow \exists d:\text{Disposition } \text{activates}(s,d) \wedge \text{manifestedBy}(d,e)$
D4	$\forall d:\text{Disposition}, e:\text{AtomicEvent}, o:\text{Object} \text{ manifestedBy}(d,e) \wedge \text{inheresIn}(d,o) \rightarrow \text{dependsOn}(e,o)$

Fig. 5 depicts a fragment of UFO-B summarizing the discussions in this section.

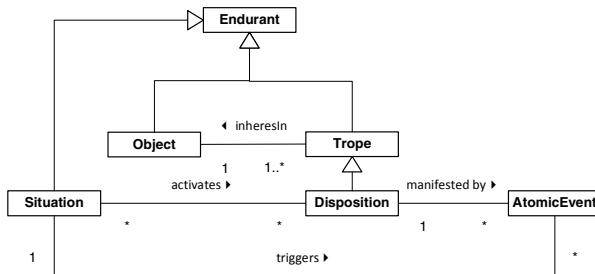


Fig. 5. Atomic events as manifestations of object dispositions

4 Evaluation and Proof of Concept

The axiomatization of UFO-B presented in this paper has been fully implemented in the formal language Alloy [12] and can be obtained from <http://nemo.inf.ufes.br/ufo-b.als>. We have used the Alloy Analyzer to show the logical consistency of the model. Moreover, in order to evaluate this model, we have employed the approach discussed in [5] of model evaluation via visual simulation. In a nutshell, we have configured a visual profile for UFO-B instances (available at <http://nemo.inf.ufes.br/ufo-b.thm>). The Alloy Analyzer then generates possible logical instances of this logical theory (given a finite domain of quantification), which are then visualized in the visual profile. By iterating through these visual instances, the modeler can detect the existence of logical models, which describe ontologically inconsistent state of affairs and, hence, rectify the ontology specification at hand.

The extended version of UFO-B reported here has been used in a number of industrial applications in the domain of media content management in a large media conglomerate in Brazil [16,17]. In that context, this ontology has been used, among other

things, in the construction of an ontology of soccer. That ontology can be used to annotate and reason with specific events taking place inside a game and to organize all the desired statistic information of a game. As reported by the organization's product owner and author of [17], UFO-B played a fundamental role in solving a number of conceptual problems of the original soccer ontology used in that organization and in the production of an ontology which is more truthful to the underlying domain.

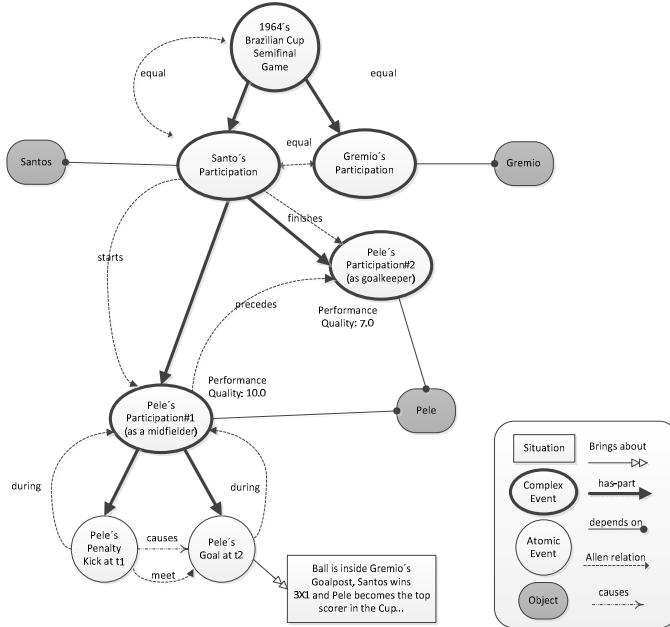


Fig. 6. A fragment of a football game described with UFO-B

The model depicted in Fig. 6 illustrates how UFO-B can be used for describing a historic soccer game. The game is the 1964's Brazilian Cup Semifinal Game between the teams Santos and Gremio, in which Pelé played both as a Midfielder and as a Goalkeeper. The game was composed of two team participations: Santos' participation and Gremio's participation. These two participations were (as always) completely synchronized (started and ended at the same time) and were also synchronized with the game itself. Pelé had two participations in the game: participation#1 (p1), as an offensive midfielder, and participation#2 (p2), as the goal keeper. (p1) begins with the beginning of Santos' participation (hence, with the beginning of the game) and stops before the end of the game (we can then infer that Pelé did not play the whole game as a Midfielder); (p2) starts in the middle of the game and lasts until the end of the game. As one can observe, there is no temporal overlap between these two participations (one *precedes* the other). In this game, Pelé's participations as midfielder and goal keeper have been predicated different levels of performance quality: as a midfielder he had a 10.0 performance mark and as a goalkeeper he had a 7.0 mark. Pelé's participation#1 is composed of other sub-events (which are part of

Santos' participation and part of the game, transitively). Two of these events are *Pelé's Penalty Kick* at a certain time t_1 , and *Pelé's scoring of a Goal* at time t_2 . These two events are causally related, i.e., it is *Pelé's Penalty Kick* that causes *Pelé's scoring of the Goal*. The situation (fact) brought about by the latter event includes changes in properties of a number of entities (e.g., Pelé's number of scores in the cup, the score of the game, the position of the ball).

5 Related Work

The notion of event is present in many upper-level ontologies (e.g., DOLCE [7], BWW [2], Kaneiwa et al [6]), as well as in lightweight semantic web ontologies such as the Event Ontology [10]. Here, due to lack of space, we can compare UFO-B only to a subset of these proposals: the Event Ontology, BWW and Kaneiwa et al.

The BWW ontology is based on the work of the philosopher Mario Bunge [2]. Over the years, it has been employed by a number of authors in the evaluation and redesign of many conceptual modeling languages and reference models. For Bunge, an event is simply defined as a formal relation between two points in the state space of an entity. Although representing an aspect of change, the change is limited to one entity, i.e., it is not the case that an event can have several participants and that individual participations can be reified, predicated upon or further mereologically decomposed. Moreover, pre- and post-states of events (involving multiple entities) cannot be modeled. In other words, one cannot model that it is the same event (e.g., a marriage) that changes the properties of both the husband and the wife. In summary, in BWW, there is also no support for an event mereology, participation differentiation, temporal relations between events and object dispositions.

The Event Ontology is used, for example, by the BBC in the design of a number of domain ontologies (e.g., the Music Ontology and the Sports Ontology³). This ontology considers events in terms of their participants, their temporal and spatial properties and their composition. The authors do not explain, however, what are the constraints applying to the relation between an event and its proper parts, i.e., no axiomatization for the assumed event mereology is presented. The temporal properties of events are defined via a mapping to *OWL Time*, which commits the model to a particular ontology of time intervals, but has the advantage of inheriting the basic axiomatization of Allen's relations. Participation of objects in events can be modeled by the *sub_event* relation together with a relation tying those objects to these subevents. This strategy, however, makes it harder to differentiate participations from other mereological parts of events. In summary, the Event Ontology has a limited treatment of mereological relations, a limited treatment of participation, a fixed treatment of temporal properties, no representation for causation and no support for the representation of dispositions and their connection to objects and events. Furthermore, except for the axiomatization of the Allen's time interval relations inherited from OWL Time, no further axiomatization for composition and participation is presented.

³ <http://www.bbc.co.uk/ontologies/>

In [6], Kaneiwa et al. propose an Upper-Level Ontology of Events. The proposed ontology offers a classification of events according to the nature of their participants, namely, between natural and artificial events. In this approach, events are defined as tuples formed by different types of constituents. For instance, natural events are characterized either by a time and location or an object, time and location. In contrast, artificial events have an agent (individual or collective), a time and a location, or an agent, an object, a time and a location. Regarding event relations, causality is defined in Kaneiwa et al. in two modes: (1) an object can cause an event, and (2) an event can cause an event. In pace with [4], we reject the former: objects do not cause events; only events cause events (sometimes via the manifestation of the disposition of these objects). The only meta-property of the causation relation put forth by Kaneiwa et al. is transitivity (in accordance with UFO-B). Regarding temporal event relations, the authors seem to consider the relations of precedes, meets, finishes and overlaps, i.e., only a subset of Allen's time interval relations. Finally, regarding parthood, the only formal constraint considered is that the proper parts of an event must be temporally included in its duration (which is also considered in UFO-B).

6 Practical Implications and Final Considerations

The view of events put forth in this paper can be summarized as follows: objects have (particularized) properties, some of which are dispositions, which are properties that are manifested in particular situations through the occurrence of events. The atomic events considered here are manifestations of single dispositions; complex events are manifestation of several dispositions. When a complex event is a manifestation of different dispositions of different objects, we can isolate those slices of such an event, which directly depend on one of these objects. We call these slices *participations*. The participation view of events is in some sense orthogonal to the mereological view. Thus, participations of individual objects in events can both be mereologically simple or complex. In this view, events (as much as objects) can also be predicated with qualitative characteristics. Events are delimited by time boundaries (here, time points which are totally ordered). Thus, events happen in time and several different temporal relations between events can be derived from the ordering of their time boundaries. These events considered here, as manifestations of dispositions, change the world, by mapping one situation to another. Situations that are brought about by the manifestation of dispositions and can activate other dispositions, making the world "tick". The unfolding of the relations between situations, dispositions and events with further activation and manifestation of other dispositions can be used to characterize an admittedly limited but very useful form of (direct and indirect) causation between events.

The work presented here is an extension of our early work on an ontological theory of events as presented in [18]. That preliminary theory of events gave us the opportunity to conduct relevant case studies for analyzing and (re)designing conceptual modeling languages, reference frameworks and domain ontologies in areas such as Enterprise Architecture, Business Process Modeling, Software Engineering, Bioin-

formatics, Telecommunications, Discrete-Event Simulation, Collaborative Processes, Service Management among others¹. The significant experience acquired on these projects in different domains over the years played an important role in selecting the type of concepts we believe a foundational ontology for the conceptual modeling of events should address. However, the ontology of events proposed in [18], despite being based on a number of results from the formal ontology literature, was presented in an informal manner. As discussed in [5,6], having an axiomatized formal semantics for a foundational ontology plays a fundamental role in supporting automated reasoning as well as the formal verification and validation of conceptual models and domain ontologies derived from it. The contributions of this paper are then two-fold: (i) firstly, it extends both the width and depth of our previous ontological treatment of events; (ii) secondly, it presents a first comprehensive axiomatization of this ontology.

Regarding (i), the work presented here extends our previous work in important ways. For instance, in [18], we mention that events can be further decomposed in sub-events. However, we make no commitment there to a specific mereology of events (extensional mereology). Moreover, in our previous work, there is no discussion on dispositions and their relation to situations and to a form of causation. This extension to the theory makes an important contribution in connecting our ontology of events with our ontology of endurants and, consequently, in supporting a systematic connection between structural and dynamic conceptual models based on these ontologies.

The notions comprising UFO-B have interesting implications to the practice of conceptual modeling. For instance, in [9], Olivé defends the representation of events in structural conceptual models. However, if events are to be represented in structural models, one should also be able to represent parthood relation between events in those models. As demonstrated in [3], parthood relations between objects can exhibit two modes of dependence, namely, generic and existential dependence. For instance, while a person depends *generically* on an instance of heart (whilst every person must have an instance heart, it does not always have to be the same instance of heart), she depends *existentially* on a particular individual instance of brain. In [3], these are called *mandatory* and *essential parts*, respectively. Now, with the adoption of an extensional mereology for events, we can show that there are no mandatory parts of events (!), i.e., all parts of events are essential parts (M9). As a consequence, whenever representing a parthood relation in a structural conceptual model (e.g., in a UML class diagram), the association end connected to the part must be deemed immutable (*readOnly*). In the same spirit, given the notion of *participation* presented here, whenever representing events in a structural conceptual model, these events should be connected to their participants via an existential dependence relation (entailing an immutability constraint in the association end connected to the types representing each participant). Moreover, the fragments of the UFO-B ontology as presented here (figures 2-5) can be re-used as *ontology design patterns* to address recurrent modeling problems. For instance, in UML class diagrams, one can represent that an event (e.g., a musical session) can be composed of zero-to-many subevents. Now, if an event can be composed of zero-to-many subevents, then it can be composed of one unique sub-event. But what then is the difference between a musical session and the unique sub-session that composes it? This situation is prevented by the (M6) axiom in UFO-B: a

musical session (business process) is either atomic or it is composed of at least two disjoint events. From a modeling perspective, this can be addressed by a direct instantiation of the model fragment in figure 2 and its associated formal constraints.

The ontological theories comprising UFO are well known and supported in the philosophical literature. For instance, our mereology of events corresponds to the standards treatment of events in the philosophical literature [13]; the notion of participation is articulated here as the perdurant counterpart of the notion of *qua individuals* (or *role instances*) [3]; the treatment of temporal ordering of events adopted here (Allen's Relations) is also well known and adopted [14]. Finally, the use of theory dispositions as tropes to articulate the relation between situations and causation also finds strong support in the formal ontology literature [11]. One of the key contributions of UFO-B is to extend a combination of existing results from formal ontology in a fuller theory for supporting the foundations of events in conceptual modeling.

It is important to highlight that what is presented in this paper is a proper fragment of the entire extended axiomatized UFO-B ontology. In particular, due to space limitations, we have left out sub-theories dealing with: (i) the differentiation of roles played by objects inside an event (the so-called *processual roles*); (ii) qualities and quality structures used to predicate qualitative aspects to events; (iii) particular aspects of events of creation, destruction and modification. These theories shall be presented in an extension of this paper. Nonetheless, the interested reader can find them in the Alloy axiomatization of UFO-B available in <http://nemo.inf.ufes.br/ufo-b.als>.

Acknowledgements. This work has been supported by FAPES (PRONEX Grant #52272362/2011). We thank Nicola Guarino for fruitful discussions and comments.

References

1. Santos Jr., P.S., et al.: An Ontology-Based Analysis and Semantics for Organizational Structure Modeling in the ARIS Method. *Information Systems*, Oxford (2013)
2. Weber, R.: *Ontological Foundations of Information Systems*. Coopers & Lybrand (1997)
3. Guizzardi, G.: *Ontological Foundations for Structural Conceptual Models*, Telematics Institute Fundamental Research Series, University of Twente, The Netherlands, vol. 15 (2005)
4. Galton, A.: States, Processes and Events, and the Ontology of Causal Relations. In: 7th International Conf. on Formal Ontology in Information Systems (FOIS 2012), Graz (2012)
5. Braga, B.F.B., et al.: Transforming OntoUML into Alloy: towards conceptual model validation using a lightweight formal method. In: ISSE, vol. 6. Springer (2010) ISSN 1614-5046
6. Kaneiwa, K., Iwazume, M., Fukuda, K.: An Upper Ontology for Event Classifications and Relations. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 394–403. Springer, Heidelberg (2007)
7. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: *Ontology Library*. WonderWeb Deliverable D18 (2003)
8. Herre, H.: General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling. In: *Handbook of Theory and Application of Ontologies*. Springer (2010)

9. Olivé, A., Raventós, R.: Modeling events as entities in object-oriented conceptual modeling languages. *Data & Knowledge Engineering* 58, 243–262 (2006)
10. Raimond, Y., Abdallah, S.: The Event Ontology (2007),
<http://motools.sourceforge.net/event/event.html>
11. Mumford, S.: Dispositions. Oxford University Press (2003)
12. Jackson, D.: Software Abstraction: Logic, Language and Analysis. MIT Press (2012)
13. Simons, P.: Parts: A Study in Ontology. Oxford University Press (1997)
14. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11) (1983)
15. Linehan, M., Barkmeyer, E., Hendryx, S.: The Date-Time Vocabulary. In: 7th International Conference on Formal Ontology in Information Systems (FOIS 2012), Graz (2012)
16. Carolo, F.P., Burlamaqui, L.: Improving Web Content Management with Semantic Technologies. In: Semantic Technology and Business Conference (SemTech 2011), San Francisco (2011)
17. Pena, R.: Semantic Support for Publishing Journalistic Content on the Web (in portuguese), MSc dissertation, Pontifical Catholic University, Rio de Janeiro, Brazil (2012)
18. Guizzardi, G., Falbo, R.A., Guizzardi, R.S.S.: Grounding software domain ontologies in the Unified Foundational Ontology (UFO): The case of the ODE software process ontology. In: 11th Iberoamerican Conference on Software Engineering, pp. 244–251 (2008)

Baquara: A Holistic Ontological Framework for Movement Analysis Using Linked Data

Renato Fileto¹, Marcelo Krüger², Nikos Pelekis³,
Yannis Theodoridis⁴, and Chiara Renso⁵

¹ PPGCC/INE - CTC, Federal University of Santa Catarina, Florianópolis, SC, Brazil
r.fileto@ufsc.br

² CTTMar, University of Itajaí Valley (UNIVALI), São José, SC, Brazil
marcelo_kruger@univali.br

³ Department of Statistics & Insurance Science,

⁴ Department of Informatics,
University of Piraeus, Greece

{npelekis,ytheod}@unipi.gr

⁵ KDDLab - IST/CNR, Pisa, Italy
chiara.renso@isti.cnr.it

Abstract. Movement understanding frequently requires further information and knowledge than what can be obtained from bare spatio-temporal traces. Despite recent progress in trajectory data management, there is still a gap between the spatio-temporal aspects and the semantics involved. This gap hinders trajectory analysis benefiting from growing collections of linked data, with well-defined and widely agreed semantics, already available on the Web. This article introduces Baquara, an ontology with rich constructs, associated with a system architecture and an approach to narrow this gap. The Baquara ontology functions as a conceptual framework for semantic enrichment of movement data with annotations based on linked data. The proposed architecture and approach reveal new possibilities for trajectory analysis, using database management systems and triple stores extended with spatial data and operators. The viability of the proposal and the expressiveness of the Baquara ontology and enabled queries are investigated in a case study using real sets of trajectories and linked data.

Keywords: Moving objects trajectories, semantic enrichment, linked data, movement analysis, ontology.

1 Introduction

Nowadays, large amounts of data about trajectories of moving objects can be gathered by using a variety of devices (e.g., smart phones equipped with GPS or just connected to a GSM network, vehicles equipped with RFID) and information systems (e.g., social Web sites that can detect changing locations of their users). A discrete raw trajectory is a time ordered sequence of spatio-temporal points $(x_1, y_1, t_1) \dots (x_n, y_n, t_n)$ ($n > 1$), where each x_i, y_i is a pair of spatial coordinates and each t_i

is an instant, representing sampled positions visited by the moving object. A sample point can be associated with keywords that carry information about the movement in that point. Lots of information can be extracted from such data, with a myriad of applications [4,8]. For instance, information extraction methods [8] can find *episodes* in raw trajectory data, i.e., maximal trajectory segments complying with a predicate [7]. However, these methods usually consider only the spatio-temporal component of trajectories, and do not address the specific content of episodes, whereas episodes can be thought of as segments of trajectories that carry specific semantics (e.g., a stop to take a picture of a monument or to take part in a sports event).

Effective analysis of movement must consider the semantics of the trajectories and the reality in which they occur. Some conceptual models have been proposed for trajectories databases [12,2]. They have introduced relevant ideas for semantic trajectories analysis, such as the concepts of stops and moves [12], later generalized to episodes [8], and dimensions for trajectories analysis [2] (e.g., goal, behavior, transportation means). Ontologies have also been proposed to support reasoning on knowledge bases describing trajectories [13,11]. However, these works do not address the automatic enrichment of trajectories with semantically precise information about specific places (e.g., restaurants, hotels, touristic spots), events (e.g., sport events, cultural events), and other relevant entities of the open dynamic world in which trajectories occur. In this article, a *semantic trajectory* is a sequence of episodes linked to specific concepts and/or instances via ontological relationships that can describe their precise semantics. Such semantic enrichment requires lots of continuously updated information, with well-defined and widely agreed semantics.

This article introduces Baquara, an ontology with an associated architecture and an approach to enable semantic enrichment and analysis of trajectories with vast and growing collections of linked data available in the Web. The proposed ontology has a rich repertoire of constructs to semantically describe trajectories and their relevant episodes with linked data. Baquara plays the role of a conceptual bridge between movement analysis and the semantic Web, by allowing movement data and associated knowledge to be connected and queried together. The proposed approach enables queries that refer to specific entities and classes taken from linked open data sources.

Our approach takes as input raw movement data associated with conventional data that may not have precise semantics (e.g., tag “Rio” may refer to a city, a state, or even a nightclub, among other possibilities). The linked data that help to describe and analyze the trajectories are selected according to the spatio-temporal scope of the movement to be analyzed, and the application domain (traffic analysis, tourism, emergency planning, etc.). Several methods can be used to find connections of movement data with linked data, including lexical and spatio-temporal matching (e.g., tag “Rio de Janeiro” associated with an episode occurring inside that city). After relevant episodes have been extracted and semantically enriched with linked data, powerful queries can be executed in the resulting knowledge base. Such queries could be from very specific, e.g., “Select the trajectories with at least one episode related to a touristic place called *Corcovado* in *Rio de Janeiro* city, even though the episode happens up to 10 kilometers away from *Corcovado*”, or abstract enough to only refer to concepts, e.g., “Select the trajectories that have a stop related to any *sport event*”.

The contributions of this article can be summarized as follows:

- An ontological framework for movement data, called Baquara, is proposed, which enables semantic enrichment and analysis of trajectories of moving objects with vast and growing collections of linked data available in the Web.
- Queries on a semantically-annotated movement data collection can be stated in our approach by using Baquara constructs in SPARQL¹ extended with spatial operators [1,6], among other language options.
- The viability of the proposed approach and the expressiveness of the enabled queries are investigated in a case study in which tagged movement data are described, according to the Baquara ontology, using linked data from DBpedia and LinkedGeoData.

The rest of this article is organized as follows. Section 2 discusses related work and contributions. Section 3 describes the proposed Baquara ontology. Section 4 presents a system architecture and a general approach for semantic enrichment of movement data with linked data. Section 5 presents a case study that illustrates the use of the proposed approach. Finally, Section 6 summarizes our contributions and future work.

2 Related Work

A pioneering work on conceptual modeling of spatio-temporal objects is MADS (Modeling Application Data with Spatio-temporal features) [9]. MADS extends the basic ER model with spatio-temporal constructs with its key premise being that spatial and temporal concepts are orthogonal. It uses the object-relationship paradigm, including the features of the ODMG (Object Database Management Group) data model, and provides spatial and temporal attributes, data types and relationships, offering a wide range of conceptual constructs to model the spatio-temporal world. A more recent contribution with focus on conceptual modeling of spatio-temporal objects changing their geographical positions but not their shapes over time comes from Spaccapietra et al. [12]. This model represents semantic trajectories as stops and moves, i.e., trajectory segments in which the object is stationary or moving, respectively. It has been the first attempt to embed semantics in the movement representation, but it lacks generality since other relevant semantic aspects are not explicitly taken into account. An extension of the “Stop-Move” model towards overcoming these limitations comes from the CONSTAnT conceptual model [2], which defines several semantic dimensions for movement analysis (e.g., goal, behavior).

Although the conceptual modeling of trajectories have seen a “convergence” to the “Stop-Move” model [8], ontologies for movement data did not find so far an agreed approach. Due to lack of space we cannot mention here all the proposals for spatio-temporal ontologies and we focus only on the ones most related to our approach. The trajectory ontology proposed by Yan et al. [13] includes three modules: the Geometric Trajectory Ontology describes the spatio-temporal features of a trajectory; the

¹ <http://www.w3.org/TR/2013/REC-sparql11-query-20130321>

Geographic Ontology describes the geographic objects; and the Domain Application Ontology describes the thematic objects of the application. These ontologies are integrated into a unique ontology that supports conjunctive queries in a traffic application. The proposal of [11] exploits a movement ontology for querying and mining trajectory data enriched with geographic and application information. Here the ontology has been used to infer application-dependent behavior from raw and mined trajectory data. Although Baquara can be seen as an extension of these approaches, it advances one important step further since it introduces rich ontological constructs, algorithms, and the use of linked data in a semi-automatic process to semantically enrich trajectories. These extensions to the widely adopted “Stop-Move” model provide a great improvement in terms of expressive power, as shown in the case study section.

3 The Baquara Ontology

The core of our approach to support semantic enrichment of movement data for trajectory analysis is the so-called Baquara² ontology. **Fig. 1** shows the high level concepts (classes) of this ontology, and the major semantic relationships between them. Each labeled rectangle represents a concept. Nesting denotes subsumption (IS-A relationship), i.e., a nested concept (e.g., Episode) is a subclass of its enclosing concept (e.g., SemanticTrace). The plus sign on the top left corner of a rectangle indicates that the respective concept is further specialized in Baquara. A dashed line between concepts denotes a semantic relationship, such as composition (PART_OF) or a specific relationship (e.g., between an Event and a Place where it occurs). Lines linking two concepts in opposite directions denote inverse relationships.

The Baquara ontology has been designed to serve as a conceptual framework for describing semantic trajectories in several application domains, ranging from urban transportation to animal ecology. The current version of Baquara has more than 100 classes and more than 200 properties. It includes all the major constructs needed for the description and analysis of trajectories. However, it can also be adapted to specific domains, if necessary, by adding class specializations and object properties.

3.1 Places, Events, and Moving Objects

Baquara ontology uses the W3C’s Time³ ontology as the Time conceptualization, and a Geometry conceptualization that is compatible with that of the OGC’s Geospatial Features⁴. Alternatives for these conceptualizations can also be considered, as standards evolve in the Web of data. These ontologies are used to define instances of Place and Event. These concepts, defined in the following, can be used to describe the spatio-temporal scope of the movement data, as well as the places and events of interest, for any particular application domain.

² The word Baquara, from the Tupi-Guarani languages, means knowledgeable, informed.

³ <http://www.w3.org/TR/owl-time>

⁴ <http://www.opengeospatial.org/standards/sfa>

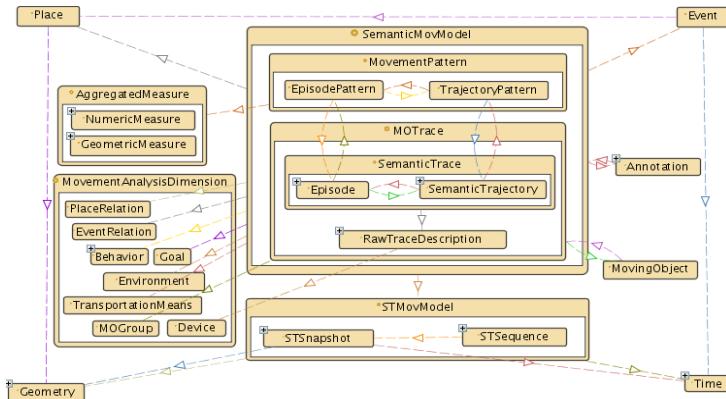


Fig. 1. An overview of the Baquara ontology

A **Place** (Definition 1) is a spatial feature relevant for movement analysis. The place's geometry, as defined by OGC's Geospatial Features, can be simple (point, line, or region) or complex (set of points, lines, or regions). Specializations of **Place** relevant for the tourism domain may, for example, include **City**, **Airport**, **Hotel**, **Restaurant**, and **LandMark**.

Definition 1. A **Place** is a tuple having a **Geometry**, and at least a name.

An **Event** (Definition 2) represents a circumstance that lasts for a time and that is relevant for movement analysis in an application domain. An event's time can be simple (instant or period) or complex (set of instants or periods), as defined by the W3C's Time ontology. Subclasses of **Event** relevant for tourism may, for example, include **Season** and **SocialEvent**. Such categories can be specialized, e.g., **SocialEvent** can be specialized to **Party**, **Meeting**, etc. Conversely, **Carnival** can be regarded as a subclass of **Party**. Instances of **Event** (and its subclasses in any abstraction level) can be related to specific instances of **Place**, as indicated by the dashed line linking these classes. Thus, an instance of city whose name is "Rio de Janeiro", can be semantically related to events occurring there, named, for example "Pan American Games of Rio 2007" and "First Strike of Rio's Public Buses in 2013". On the other hand, events like "Christmas holidays" may not be associated with any particular place, because they occur in many places.

Definition 2. An **Event** is a tuple having a **Time**, at least a name, and zero or more relations with places.

A **MovingObject** (Definition 3) is anything that moves and that can be distinguished from other moving objects through its **MOid** (moving object identifier). In Baquara, a **MovingObject** can be a **Person**, a **Vehicle**, an **Animal**, or even a **Storm** (that can be represented as a moving region).

Definition 3. A **MovingObject** is a tuple having an **MOid**, and collections of **RawTraceDescription** and **SemanticTrajectory** instances.

The key constituents of the above definition, `RawTraceDescription` and, more important, `SemanticTrajectory`, are presented in detail in the following.

3.2 Movement Models and Description Statements

The entities used to represent movement in Baquara are subclasses of `STMovModel` and `SemanticMovModel`. A `STMovModel` (Spatio-temporal Model of Movement) just describes spatio-temporal representations of movement. It can refer to raw traces, processed traces (e.g., resulting of data cleansing and map matching), or even semantic trajectories and trajectory patterns having spatio-temporal counterparts. Such data are efficiently managed by moving object data management systems, such as *Hermes* [10]. The abstract class `STMovModel` has two concrete subclasses, `STSsnapshot` and `STSequence`. A `STSsnapshot` refer to a particular spatio-temporal situation that is represented by a `Geometry` and a `Time`. A `STSequence` is a sequence of instances of `STSsnapshot` ordered according to their non-overlapping values of time. It is worth to note that the Baquara ontology does not restrict the geometries of moving objects. Baquara allows moving objects to be points, lines, or regions (e.g., storms). However, most case studies and moving object data management systems focus on point geometries.

A `SemanticMovModel` can be a `MOTrace`, or a `MovementPattern`. A `MOTrace` (Moving Object Trace) is an abstract class that describes movement done by a specific `MovingObject`. It can be a `RawTraceDescription` of raw data collected by a `Device`, or a `SemanticTrace` built by post-processing raw data.

A `RawTraceDescription` (Definition 4) is used to semantically describe raw data about movement. For example, to indicate the sensor device used to collect the trace, and other relevant information, such as the spatio-temporal precision, and the sampling rate. The spatio-temporal representation of an episode may differ from that of raw data, due to necessary transformations, including data cleansing, interpolation between samples, and map matching. Therefore, each class, `Episode` and `RawTraceDescription`, has its own `STMovModel`.

Definition 4. A `RawTraceDescription` is a tuple with one `STMovModel`, and an indication of the device used to collect that spatio-temporal data.

A `SemanticTrace` is an abstract class whose concrete subclasses are `SemanticTrajectory` (Definition 5) and `Episode` (Definition 6). An `Episode` is any noticeable happening in a trajectory segment, such as a stop or a move, that can be detected by a trajectory segmentation process; for instance, a stop may be detected according to whether the segment exceeds or not some threshold values on movement predicates (area covered, speed, etc.). A `SemanticTrajectory` is a time ordered sequence of episodes.

Definition 5. A `SemanticTrajectory` is a tuple having an `id` and a time ordered sequence of `Episode` instances.

Definition 6. An `Episode` is a tuple with the `RawTraceDescription` of the raw data used to build it, and, optionally, one `STMovModel`.

Differently from a MOTrace, a MovementPattern describes a conceptual movement that is not associated with a specific MovingObject. It can happen or not in some movement database (e.g., a movement starting at home and ending at work). A MovementPattern is an abstract class that generalizes TrajectoryPattern (Definition 7) and EpisodePattern (Definition 8).

Definition 7. A **TrajectoryPattern** is a tuple with a time ordered sequence of EpisodePattern instances, and arbitrary numbers of complying semantic trajectories, and aggregated measures about these trajectories.

Definition 8. An **EpisodePattern** is a tuple with arbitrary numbers of complying episodes, and aggregated measures about these episodes.

A MovementPattern (either TrajectoryPattern or EpisodePattern) refers to its compliant SemanticTraces (SemanticTrajectory or Episode collections, respectively), i.e., the ones with compatible traits (e.g., trajectories whose first episode is a stop at a Market). A MovementPattern can also have aggregate measures of its compliant SemanticTrace collections. These measures can be numeric (e.g., the total distance traveled and the time spent by all compliant traces) or geometric (e.g., an aggregate geometry representing the compliant traces).

Finally, description statements (Definition 9) allow the semantic annotation of the previously described entities of the SemanticMovModel with linked data.

Definition 9. A **description statement** for an instance r of a class R is a triple $DS(r, P, V)$ where P refers to a property defined for instances of R , and V is a value that can be a typed literal (string or number), an instance of a class, or a class itself.

Any instance r of a class R subsumed by SemanticMovModel can have an arbitrary number of description statements $DS(r, P, V)$. The general property hasAnnotation can be used for general descriptions. For example, an Episode can be related to an annotation having the value “bus”. However, such an annotation does not have specific semantics; it does not specify, for instance, if the mentioned “bus” plays the role of transport means or it is just something that captured the interest of the moving object in the annotated episode.

Other properties and classes are predefined in Baquara for making description statements with specific semantics. For example, an Episode can alternatively be described with the predefined property usesTransportationMeans pointing to e.g., the class Bus subsumed by class TransportationMeans. The values of specific properties can be taken from SemanticAnalysisDimensions, i.e., hierarchies of semantically related concepts or instances. Examples of such dimensions also include Goal and Behavior, among others, defined in [2]. Baquara defines specific properties for those and other analysis dimensions, such as MOGroup (Moving Objects Group), EventRelation, and PlaceRelation. The former allows for movement analysis according to general traits of moving objects (e.g., their classes, such as Vehicle and Person), without explicitly identifying them. The latter describe the kinds of relations that a MovementModel (e.g., an Episode) can have with an Event or a Place, respectively. For instance, an Episode may happen in a particular Place during an Event. Alternatively, an Episode can

happen when the moving object just observes a Place (maybe from distance) or prepares for taking part in an Event. Thus, distinct properties can link a MovementModel to a Place (e.g., the city “Rio de Janeiro”) or an Event (e.g., the sports event “Pan American Games of Rio 2007”).

4 Semantically Enriching Trajectories with Linked Data

The ontology described in Section 4 is a conceptual framework to support semantic enrichment and analysis of movement data. This section presents the data enrichment process based on that ontology. This process allows arbitrary techniques for information extraction from movement data (e.g., to find trajectory episodes). It exploits ontologies and linked data to delineate movement analysis dimensions with well-defined semantics, enriching the movement analysis possibilities.

4.1 Problem Description

Consider that movement data (MoD) is provided as a relation of the form:

$$\text{MoD } (\underline{\text{id}}, \text{MOid(fk)}, \text{T}, \text{S}, \text{A}_1, \dots, \text{A}_n) \quad (1)$$

where id is the tuple identifier (primary key), MOid is the moving object identifier (foreign key), T is a validity time (instant or period) when the moving object had the position and shape represented by S , S is a geometry (point, line, region), and $\text{A}_1, \dots, \text{A}_n$ ($n \geq 0$) are descriptive attributes (e.g., *TransportationMeans*, *Tag*, *Goal*). Note that such a relation can hold relevant episodes instead of raw data.

A MoD relation can carry lots of information. However, this spatio-temporal and descriptive information (when the latter is available) without links to knowledge about the geographic space, relevant events, possible goals, and transportation means, among other issues, may be not enough to explain movement and support intelligent analysis. Thus, a semantically rich model of movement, such as that of Baquara described in Section 3, must be built from the MoD. The challenge is to build such a model for large MoD relations, taking into account potentially huge amounts of information and knowledge about the relevant data analysis dimensions, which may vary according to the application domain. This problem can be divided in three tasks:

- Task 1. *extract episodes* from raw MoD;
- Task 2. *find connections* (e.g., *lexical and spatial*) between episodes and information about the environment in which the episodes occur; and
- Task 3. *devise proper semantic relations* between episodes and environment information to build description statements that can support analyses.

When large amounts of data are involved, automated methods are necessary to solve each one of these tasks. A variety of methods are currently available to solve Task 1, mainly by processing spatio-temporal data about movement [8]. Methods to solve Tasks 2 and 3 are still open issues, particularly if the attributes in the MoD are too generic (e.g. *Tag*) to be directly mapped to specific analysis dimensions via

specific properties. On the other hand, we argue that the vast and growing collections of linked data available in the semantic Web can supply the information needed to realize these tasks in many cases. Furthermore, there is a potential for Task 1 to benefit from descriptive attribute values with the well-defined semantics of linked data too.

4.2 Using Linked Data in the Semantic Enrichment Process

The semantic enrichment process proposed in this work takes MoD in the form of (1), along with ontologies and linked data from various sources, to semantically describe movement in accordance to the Baquara ontology described in Section 3. The Baquara ontology provides a conceptual model to represent trajectories, episodes, patterns, and other constructs referring to movement. Its concepts and properties allow the description of movement using linked data.

Fig. 2 illustrates a system architecture to support the proposed semantic enrichment process. This process starts by loading the Baquara ontology in the knowledge management system. Then, linked data with the same spatio-temporal scope as the MoD to be analyzed can be selected from several sources (e.g., by using their SPARQL endpoints or REST APIs). The collected knowledge, initially represented as RDF triples, is used in the semantic enrichment, warehousing, and mining of the MoD.

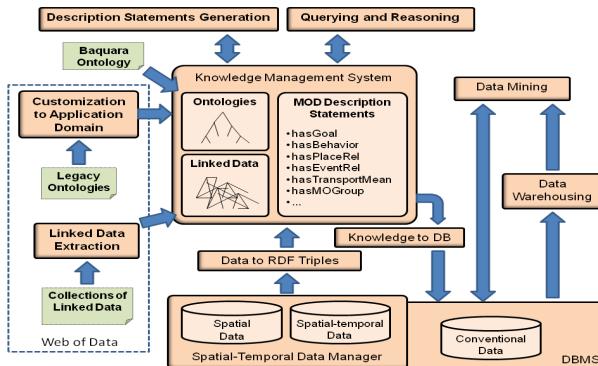


Fig. 2. The proposed architecture for semantic enrichment and analysis of MoD

The description statements generation takes MoD and linked data to derive description statements. This process can be interlaced with the extraction of relevant episodes of the raw MoD contained in the spatio-temporal database. MoD can be converted into RDF triples, to process as knowledge when convenient. Conversely, the RDF triples of the resulting knowledge base with description statements based on linked data can be converted into a format that allows direct processing in a conventional or spatio-temporal DBMS, for efficiency data warehousing and mining.

Algorithm 1 outlines a general method for automatically generating description statements. It takes as input *MovData*, i.e. a MoD relation (as described in Section 4.1), the Baquara ontology, and linked data with the same spatio-temporal scope as *MovData* (e.g., whose valid time and spatial extents are inside the minimum bounding box inclosing the *MovData*), for a particular application domain (e.g., tourism).

It returns as output DS, a collection of description statements. The latter describes movement sample points or episodes, by connecting them to linked data via properties predefined or specialized in the Baquara ontology.

```

INPUT:   MovData(id,MOid,T,S,A1, ...,An) % Movement data
        LD    % Baquara ontology and collected linked data
OUTPUT:  DS(id,property,value)    % Description statements
1.   DS =  $\emptyset$ 
2.   FOR EACH r IN MovData DO
3.       Matches = FindMatches(r,LD);
4.       FOR EACH v IN Matches DO
5.           CandidateProperties = ChooseProperties(r,v);
6.           FOR EACH p IN CandidateProperties DO
7.               DS = DS + (r.id,p,v);
8.   RETURN DS

```

Algorithm 1. General approach for generating description statements

Initially, the set of generated description statements (DS) is empty (line 1). Then, for each tuple $r \in \text{MovData}$, the automatic method `FindMatches(r, LD)` finds its matches with linked data in LD (lines 2 and 3). A variety of methods can be used for this purpose, such as spatial proximity and lexical similarity between conventional attribute values of the MoD and labels of linked data entities (e.g., the tag “Rio” and the labels of entities in linked data collections). Of course, problems can arise when looking for matches, such as ambiguities. Additional information (e.g., classes of the moving objects and linked data resources) can help solve these problems. Likewise, such information can help `ChooseProperties(r, v)` find properties to generate description statements for each tuple r and linked data resource v (lines 4 to 7).

The current version of a prototype built to evaluate the proposed approach for MoD enrichment and analysis with linked data employs well-known spatio-temporal and text matching techniques [5,3,14] to find matches of MoD with linked data. It uses either the general `hasAnnotation` or the `isAt` (a place) property to generate description statements. The investigation of more sophisticated methods to generate specific description statements is theme for future work.

5 Case Study

We investigate the viability of our approach to semantically enrich and analyze MoD through a case study. In particular, we take raw trajectories and linked data about Brazil available at the Web, extract episodes, generate description statements, and evaluate some example queries in the resulting knowledge base, as described below.

5.1 Movement Data

The relation `RawFlickTrajsBrazil(id, MOid, Instant, Lat, Long, Tag)` is the MoD used in our case study. It was extracted from CoPhIR⁵, a collection of data

⁵ <http://cophir.isti.cnr.it>

about images uploaded in Flickr⁶. The spatial coordinates (`Lat`, `Long`) refer to the positions indicated by the Flickr users when uploading their pictures. The `Instant` of each sample was collected by the devices used to take the pictures. Though the time of the devices may not be set correctly, it can be used to calculate the time intervals between consecutive sample points. We have verified by visual inspection that the coordinates may refer to the user's position when taking the picture or to a pictured object itself.

After extracting tuples with spatio-temporal points inside Brazil, we separated the trajectories as time ordered sequence points visited by each user during each day and, then, we eliminated trajectories with segments having speed higher than 500 km/h (i.e., following a simple trajectory reconstruction technique; evaluating more sophisticated approaches for trajectory reconstruction is beyond the scope of this article). The resulting raw data collection has 2143 trajectories owned by 564 distinct users, and consisting of 14504 sampled positions. These positions are associated with 12443 distinct tags. The total number of tuples in `RawFlickTrajsBrazil` is 117146, i.e., each spatio-temporal sample point is associated to 8.08 tags in average. We have extracted 971 stops from these trajectories. Each of these stops corresponds to a period of at least 30 minutes without moving more than 500 meters. These stops are associated to 6278 distinct tags, in a total of 45768 (stop,tag) pairs, i.e., around 47 different tag values associated to each stop, in average. After cleaning irrelevant tag values, we used `Triplify`⁷ to transform the MoD from relations into RDF triples, in accordance to the conceptual model defined by the Baquara ontology.

5.2 Semantic Enrichment with Linked Data

We have accessed triple sets available on the Web pages, SPARQL endpoints, and REST endpoints to extract subsets of linked data from DBpedia⁸ (a knowledge base built from Wikipedia⁹), LinkedGeoData¹⁰ (a large collection of geographic entities' descriptions taken from OpenStreetMap¹¹), and GeoCodes¹² (another knowledge base about geographic entities). The extracted subsets are compatible with the spatio-temporal scope of the raw MoD considered for analyses, i.e., referring to Brazil and/or the time period between 2007 and 2008. The extracted linked data was stored in the triple repository Virtuoso¹³, which supports spatial data extensions and GeoSPARQL¹⁴ [1] queries with spatial operators.

Then we have investigated matches between labels of a selected collection of linked data and the tags associated with the sample points and episodes of our trajectory data collection. We have used just approximate string matching methods and

⁶ <http://www.flickr.com>

⁷ <http://triplify.org>

⁸ <http://dbpedia.org>

⁹ <http://www.wikipedia.org>

¹⁰ <http://linkedgeodata.org>

¹¹ <http://www.openstreetmap.org>

¹² <http://www.geocode.com>

¹³ <http://virtuoso.openlinksw.com>

¹⁴ <http://www.opengeospatial.org/standards/geosparql>

geographic coordinates to help disambiguate in some cases. Most of the matches found are tag values from CoPhIR trajectories matching labels of entities classified as different kinds of places (`PopulatedPlace`, `NaturalPlace`, `Amenity`, etc.) in the collected linked data. We have also found a smaller number of matches with specific kinds of events (`SportsEvent`, `SoccerTournament`, `FootballMatch`, etc.) and organizations (`SportsTeam`, `SoccerClub`, `SambaSchool`, etc.). This semantic enrichment process enabled us to generate some description statements to answer queries such as the ones presented in the following.

5.3 Analytical Queries

There follows some examples of GeoSPARQL queries that can be executed in the knowledge base resulting from the semantic enrichment of our collection of Flickr trajectory data. They use the `bq` namespace to refer to the Baquara ontology.

Query 1 (single episode query): Select trajectories with at least one episode that mentions and occurs up to 10 km from Corcovado (the mountain of Rio de Janeiro in whose top stands the statue of Christ the Redeemer).

```
SELECT ?trajectory WHERE {
  ?trajectory a bq:SemanticTrajectory;
  bq:hasEpisode ?episode.
  ?episode bq:hasAnnotation ?a. ?a bq:hasValue ?v.
  ?v a <http://dbpedia.org/ontology/Mountain>;
  rdfs:label "Corcovado"@pt; geo:geometry ?cGeo.
  ?rio a <http://dbpedia.org/ontology/City>;
  rdfs:label "Rio de Janeiro"@pt; geo:geometry ?rioGeo.
  FILTER(bif:st_intersects (?cGeo,?rioGeo,20) &&
         bif:st_intersects (?eGeo,?cGeo,10)) }
```

Query 2 (multiple episodes query): Select trajectories with a stop in an `Amenity`, a stop mentioning a `SportsEvent`, and a stop lexically related to "Beach".

```
SELECT ?trajectory WHERE {
  ?trajectory a bq:SemanticTrajectory;
  bq:hasEpisode ?s1, ?s2, ?s3.
  ?s1 a bq:Stop; bq:occursIn ?p1.
  ?s2 a bq:Stop; bq:hasAnnotation ?a2.
  ?s3 a bq:Stop; bq:hasAnnotation ?a3.
  ?a2 bq:hasValue ?v2. ?a3 bq:hasValue ?v3.
  ?p1 a <http://linkedgeodata.org/ontology/Amenity>.
  ?v2 a <http://dbpedia.org/resource/SportsEvent>.
  FILTER(regex(?v3,"Beach")) }
```

Commenting on the above queries, the raw trajectories extracted from Flickr have sample points annotated with the tag "Corcovado" that are far away from that

mountain, probably because it can be seen and pictured from many positions in Rio. The FILTER clause in Query 1 ensures that the considered mountain labeled with "Corcovado"@pt is the one inside the city called Rio de Janeiro and that the stop mentioning that mountain occurs up to 10 km from it. Further, in Query 2, the bidding of the description statements of the stops s1 and s2 with values from specific classes from LinkedGeoData and DBpedia, respectively, are more precise than just matching strings, as it is done in the filter condition with v3.

6 Conclusions and Future Work

Vast collections of linked data about real world entities and events have been fed and continuously updated on the Web. However, their potential to leverage movement understanding has not been exploited yet. This article gives the following contributions towards using linked data to help movement analyses: (i) an ontology, called Baquara, for semantic trajectories enrichment with linked data; (ii) an architecture to narrow the gap between trajectory mining and the semantic Web; (iii) an automated method to derive semantic annotations from movement data with free annotations; (iv) examples of analytical queries enabled by this proposal through a case study with real data available at the Web. Though the queries presented in this article run on triple stores, the latter can be used just as means to handle knowledge. After semantic enrichment, the resulting knowledge can be converted into conventional and spatio-temporal databases, for more efficient analysis and mining.

In our future work we plan to: (i) evaluate the proposed approach with spatio-temporal and linked data from distinct domains; (ii) develop efficient methods to derive precise description statements from different data collections; and (iii) investigate the use of linked data for trajectories warehousing and trajectories mining.

Acknowledgments. This work has been supported by the European Union's IRSES-SEEK (grant 295179) and FP7-DATASIM (grant 270833) projects, and Brazil's CNPq (grant 478634/2011-0) project. Nikos Pelekis' research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales.

References

1. Battle, M., Kolas, D.: Enabling the Geospatial Semantic Web with Parliament and GeoSPARQL. *Semantic Web Journal* 3(4), 355–370 (2012)
2. Bogorny, V., Renso, C., Aquino, A., Siqueira, F.L., Alvares, L.O.: CONSTAnT - A Conceptual Data Model for Semantic Trajectories of Moving Objects. *Transactions in GIS* 8(2) (2013)
3. Cohen, W.W., Ravikumar, P.D., Fienberg, S.E.: A Comparison of String Distance Metrics for Name-Matching Tasks. In: IIWeb, pp. 73–78 (2003)

4. Giannotti, F., Nanni, M., Pedreschi, D., Pinelli, F., Renso, C., Rinzivillo, S., Trasarti, R.: Unveiling the complexity of human movement by querying and mining massive trajectory data. *The VLDB Journal* 20(5), 695–719 (2011)
5. Navarro, G.: A guided tour to approximate string matching. *ACM Computing Surveys* 33(1), 31–88 (2001)
6. Kyzirakos, K., Karpathiotakis, M., Koubarakis, M.: Strabon: A Semantic Geospatial DBMS. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 295–311. Springer, Heidelberg (2012)
7. Mountain, D., Raper, J.F.: Modelling human spatio-temporal behaviour: a challenge for location-based services. In: 6th Int. Conf. on GeoComputation, Brisbane, Australia, pp. 24–26 (2001)
8. Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M.L., Gkoulalas-divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., Yan, Z.: Semantic trajectories modeling and analysis. *ACM Computing Surveys* 45 (2013)
9. Parent, C., Spaccapietra, S., Zimányi, E.: Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach. Springer (2006)
10. Pelekis, N., Frentzos, E., Giatrakos, N., Theodoridis, Y.: HERMES: aggregative LBS via a trajectory DB engine. In: SIGMOD Conf., pp. 1255–1258 (2008)
11. Renso, C., Baglioni, M., de Macedo, J.A.F., Trasarti, R., Wachowicz, M.: How you move reveals who you are: understanding human behavior by analyzing trajectory data. *Knowledge and Information System Journal (KAIS)*, 1–32 (June 2012)
12. Spaccapietra, S., Parent, C., Damiani, M.L., de Macedo, J.A., Porto, F., Vangenot, C.: A conceptual view on trajectories. *Data and Knowledge Engineering* 65(1), 126–146 (2008)
13. Yan, Z., Macedo, J., Parent, C., Spaccapietra, S.: Trajectory Ontologies and Queries. *Transactions in GIS*, 12(suppl. 1), 75–91 (2008)
14. Yao, B., Li, F., Hadjieleftheriou, M., Hou, K.: Approximate string search in spatial databases. In: Int. IEEE Conf. on Data Engineering (ICDE), pp. 545–556 (2010)

From Structure-Based to Semantics-Based: Towards Effective XML Keyword Search

Thuy Ngoc Le¹, Huayu Wu², Tok Wang Ling¹, Luochen Li¹, and Jiaheng Lu³

¹ National University of Singapore

{ltngoc, lingtw, luochen}@comp.nus.edu.sg

² Institute for Infocomm Research, Singapore

huwu@i2r.a-star.edu.sg

³ Renmin University of China

jiahenglu@ruc.edu.cn

Abstract. Existing XML keyword search approaches can be categorized into tree-based search and graph-based search. Both of them are structure-based search because they mainly rely on the exploration of the structural features of document. Those structure-based approaches cannot fully exploit hidden semantics in XML document. This causes serious problems in processing some class of keyword queries. In this paper, we thoroughly point out mismatches between answers returned by structure-based search and the expectations of common users. Through detailed analysis of these mismatches, we show the importance of semantics in XML keyword search and propose a semantics-based approach to process XML keyword queries. Particularly, we propose to use Object Relationship (OR) graph, which fully captures semantics of object, relationship and attribute, to represent XML document and we develop algorithms based on the OR graph to return more comprehensive answers. Experimental results show that our proposed semantics-based approach can resolve the problems of the structure-based search, and significantly improve both the effectiveness and efficiency.

Keywords: XML, keyword search, object, semantics.

1 Introduction

Current approaches for XML keyword search are structure-based because they mainly rely on the exploration of the structure of XML data. They can be classified into the tree-based and the graph-based search. The tree-based search is used when an XML document is modeled as a tree, i.e. without ID References (IDREFs) such as [22,17,12,23,20], while the graph-based search is used for XML documents with IDREFs such as [7,9,3,13,10]. Due to the high dependence on hierarchical structure and unawareness of real semantics in XML data, these approaches suffer from several serious limitations as illustrated in the following example.

Consider an XML keyword query $Q = \{\text{Bill}, \text{John}\}$ issued to the XML data in Fig. 1, in which the query keywords match first name of two students. Let us discuss answers for this query returned by the LCA-based (Lowest Common Ancestor) approach, a representative of the tree-based search. The LCA-based approach returns the document root as an answer for Q , which is intuitively meaningless for users. Suppose

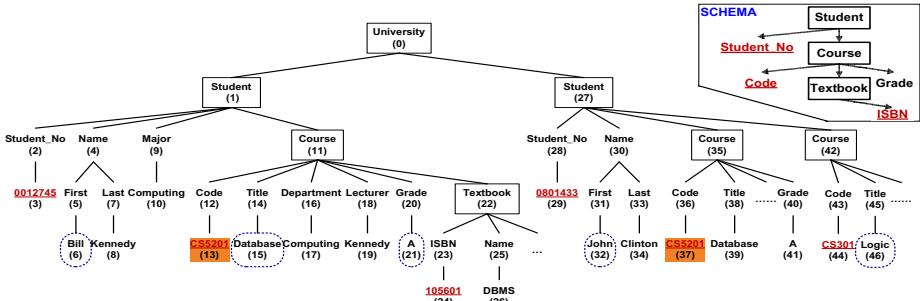


Fig. 1. university.xml

we could tell that two objects are the same if they belong to the same object class and have the same object identifier (ID) value. Then Course_(11) and Course_(35) refer to the same object <Course:CS5201> because they belong to the same object class Course and have the same object ID value CS5201. <Course:CS5201> is the common course taken by both students Bill and John and should be an answer. LCA-based approaches miss this answer because of unawareness of object, object ID and duplication of the same object. Thus, the common courses taken by both students are not recognized. In fact, other approaches of the tree-based and graph-based search suffer from similar problems, which will be demonstrated in Sec. 3 and 4.

The question we consider in this paper is that besides values, what benefits we can derive from paying attention to the role of tags in XML document, particularly for the problem of keyword queries. For this purpose, we introduce ORA-semantics and exploit it for XML keyword search. ORA-semantics stands for semantics of *Object*, *Relationship* and *Attribute* derived from XML tags. Once an XML document is defined with ORA-semantics, we can develop an effective semantics-based approach for XML keyword search which can solve limitations of the structure-based search.

In brief, the contributions of our work are as follows.

- We illustrate the limitations of both types of the structure-based search (the tree-based and the graph-based) in XML keyword search in details (Sec. 3 and 4).
- We introduce ORA-semantics and show that ORA-semantics plays an important role to effectively process XML keyword queries. Thus we propose semantics-based approach for XML keyword search, in which we use OR graph, which can capture full ORA-semantics, to represent XML document (Sec. 5 and 6).
- We perform comprehensive experiments to compare our semantic-based approach with the structure-based approaches (including XKSearch [22] and BLINK [7]). Experimental result shows the significant superiority of our methods because it can solve limitations of the structure-based search efficiently (Sec. 6).

2 Terminology

This section presents concepts of object, relationship and attribute used in this paper and uses the XML data in Fig. 1 for illustrating examples.

Concept 1 (Object). In an XML data tree, an object is represented by a group of nodes, starting at a tag w.r.t. object class, followed by a set of attributes and their associated values. Each object belongs to an object class and has a unique object ID value, which can be single or composite.

Concept 2 (Object node vs. non-object node). Among nodes describe an object, the one w.r.t. an object class is called object node and all remaining nodes are called non-object nodes. Each non-object node is associated with a corresponding object node.

For example, `Student_(1)` is an object node whereas `Name_(4)` is a non-object node belonging to object node `Student_(1)`.

Concept 3 (The same object). Two objects are the same if they belong to the same object class and have the same object ID value.

For example, `Course_(11)` and `Course_(35)` refer to the same object `Course`: `CS5201` because they belong to the same object class `Course` and they have the same object ID value `CS5201`. Object nodes which refer to the same object are usually identical. If we find two nodes that are not identical, they are still considered as referring to the same object as long as they are of the same object class and have the object ID.

Concept 4 (Relationship). Objects may be connected through some relationship which can be explicit or implicit. An explicit relationship explicitly appears in an XML data as a node, whereas an implicit relationship is reflected by the connection among objects.

For example, in Fig. 1, there is no explicit relationship but several implicit relationships such as relationships between `Student_(1)` and `Course_(11)`.

Concept 5 (Attribute). An attribute can be an object attribute or a relationship attribute. In XML data, it can be a child of an object node, a child of an explicit relationship node or a child of the lowest object of an implicit relationship node.

For example, `Lecturer` is an attribute of object class `Course` whereas `Grade` is a relationship attribute of an implicit relationship between a `Student` and a `Course`.

3 Revisiting the Tree-Based XML Keyword Search

Since almost all tree-based approaches are based on LCA semantics such as SLCA [22], VLCA [12], ELCA [23], we use the LCA-based approach as a representative of the tree-based search onward. In this section, we systematically point out limitations of the LCA semantics by comparing answers returned by the LCA semantics and answers that are probably expected by users. We use the XML data in Fig. 1 for illustration. It is worthy to note that `Course_(11)` and `Course_(35)` refer to the same object `<Course:CS5201>` despite of appearing as different nodes.

3.1 Meaningless Answer

Example 3.1. $Q_{3.1} = \{\text{Bill}\}$.

LCA answer. The LCA-based approach returns node `Bill_(6)`. However, this is not useful since it does not provide any supplementary information about `Bill`. This happens when a returned node is a non-object node, e.g., an attribute or a value.

Reason. The LCA-based approach cannot differentiate object and non-object nodes. Returning object node is meaningful whereas returning non-object node is not.

Expected answer. The expected answer should be forced up to `Student_(1)`, the object w.r.t. to `Bill_(6)` since it contains supplementary information related to `Bill`.

3.2 Missing Answer

Example 3.2. $Q_{3.2} = \{\text{Bill, John}\}$.

LCA answer. The LCA-based approach returns the document root which is definitely not meaningful.

Reason. Due to unawareness of semantics of object, the LCA-based approach can never recognize that, `Course_(11)` and `Course_(35)` refer to the same object `Course CS5201`. This is the common course taken by the two students Bill and John.

Expected answer. The expected answer should be the common course taken by these two students, i.e., `<Course:CS5201>` appearing as `Course_(11)` and `Course_(35)`.

3.3 Duplicated Answer

Example 3.3. $Q_{3.3} = \{\text{CS5201, Database}\}$.

LCA answer. Two answers `Course_(11)` and `Course_(35)` of this query are duplicated because the two nodes refer to the same object `<Course:CS5201>`.

Reason. Similar to Example 3.1, this problem is caused by the unawareness of duplication of object having multiple occurrences.

Expected answer. Either of `Course_(11)` or `Course_(35)` should be returned, but not both since they are different occurrences of the same object `<Course:CS5201>`.

3.4 Problems Related to Relationships

Example 3.4. $Q_{3.4} = \{\text{Database, A}\}$.

LCA answer. The LCA-based approach returns `Course_(11)` and `Course_(35)` as answers. These answers are incomplete because ‘A’ grade is not an attribute of a course, but it is grade of a student taking the course instead. On the other hand, `Grade` is a relationship attribute between `Student` and `Course`, not an object attribute.

Reason. The LCA-based approach cannot distinguish between an object attribute and a relationship attribute under an object node.

Expected answer. The proper answer should be all students taking course `Database` and getting an ‘A’ grade. To do that, the answer should be moved up to contain other objects (e.g., students) participating in the relationship that ‘A’ grade belongs to.

3.5 Schema Dependence

There may be several designs for the same data source. The XML data in Fig. 1 can be represented by another design as in Fig. 2 with different hierarchical structure among object classes, e.g., `Course` becomes the parent of `Student`.

Example 3.5. $Q_{3.5} = \{\text{Bill, Database}\}$.

LCA Answer. With the design in Fig. 1, the LCA-based approach returns `Student_(1)`. With the design in Fig. 2, `Course_(1)` is returned. As shown, answers for different designs are different though these designs refer to exactly the same information and we are dealing with the same query.

Answers of other queries related to more than one object also depend on XML hierarchical structure. Let us recall $Q_{3.2} = \{\text{Bill, John}\}$, $Q_{3.4} = \{\text{Database, A}\}$ and discuss the answers from the design in Fig. 2 for these queries. For $Q_{3.2}$, `Course_(1)` is an answer. For $Q_{3.4}$, `Course_(1)` is also returned. Compared with the answers the root for $Q_{3.2}$ and `Course_(11)` and `Course_(35)` (without students as their children) for $Q_{3.4}$ from the design in Fig. 1, the ones from the design in Fig. 2 are different.

Reason. Answers from the LCA semantics rely on the hierarchical structure of XML data. Different hierarchical structures may provide different answers for the same query.

Expected Answer. Users issue a keyword query without knowledge about the underlying structure of the data. Thus, their expectation about the answers is independent to the schema design. Therefore, the expected answers should also be semantically the same with all designs of the same data source.

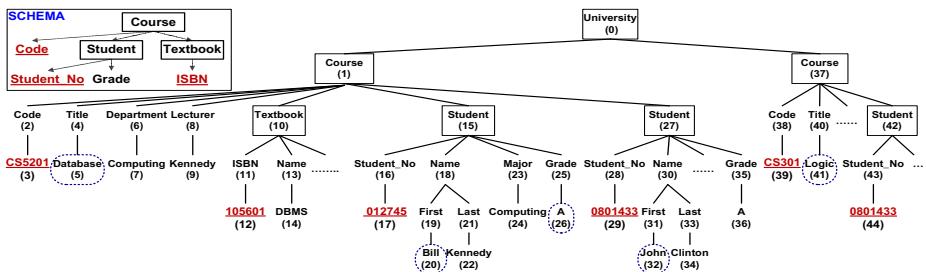


Fig. 2. Another design for the XML data in Fig. 1

Summary. The main reasons of the above problems are the high dependence of answers returned by the LCA-based search on the hierarchical structure of XML data (e.g., $Q_{3.5}$, $Q_{3.2}$, $Q_{3.4}$), and the unawareness of semantics of object, relationship and attribute. Particularly, unawareness of objects causes *missing answers* (e.g., $Q_{3.2}$), and *duplicated answer* (e.g., $Q_{3.3}$) because the LCA-based approach cannot discover the same object. Unawareness of object and attribute cause *meaningless answer* (e.g., $Q_{3.1}$) because it cannot differentiate XML elements. Unawareness of relationship and attribute cause the *problems related to relationship* (e.g., $Q_{3.4}$) because of it is unable to differentiate an object attribute and a relationship attribute.

4 Revisiting the Graph-Based XML Keyword Search

The graph-based search can be applied for both XML tree (without IDREF) and XML graph (with IDREFs). In the absence of IDREF, the graph-based search suffers from

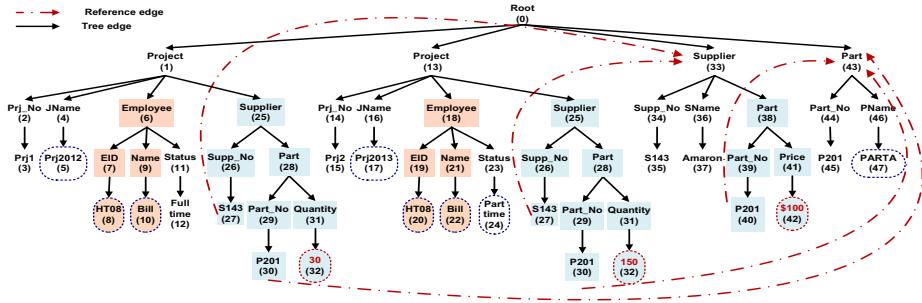


Fig. 3. An XML document with IDREFs

the same problems as the LCA-based search does. With IDREFs, object and object ID are observed and utilized. An object can be referenced by an IDREF, which has the same value with its object ID to avoid duplication. This helps the graph-based search handle some but not all problems of the LCA-based search. Particularly, the *problems related to relationship* and *meaningless answers* cannot be solved no matter IDREF is used or not. The other problems including *missing answer*, *duplicated answer* and *schema dependence* can be solved if the ID Reference mechanism applies to all objects. Otherwise (existing some objects without IDREF), they cannot be solved totally.

For generality, in this section, we use the XML data in Fig. 3 which contains both objects with and without IDREF to illustrate problems of the graph-based search. We apply the widely accepted semantics *minimum Steiner tree* [5,6] for illustrating the problems. In the XML data in Fig. 3, Object <Employee:HT08> is duplicated with two occurrences Employee_(6) and Employee_(26). Ternary relationship type among Supplier, Project and Part means suppliers supply parts to projects. Quantity is an attribute of this ternary relationship and represents the quantity of a part supplied to a project by a supplier. Besides, binary relationship between Supplier and Part has an attribute Price to represent the price of a part supplied by a supplier.

4.1 Problems Cannot Be Solved with IDREF

IDREF mechanism is aware of semantics of object and object ID. However, the semantics of relationship and attribute is still not recognized and utilized which causes the problems of *meaningless answer*, and *problems related to relationship*.

Meaningless answer. Not differentiating object and non-object nodes cause meaningless answer when the returned node is a non-object node. For example, for $Q_{4.1} = \{\text{Amazon}\}$, the answer is only Amazon_(45) without any other information.

Problems related to relationships. Without semantics of relationship, the graph-based search cannot distinguish object attribute and relationship attribute, and cannot recognize n-ary ($n \geq 3$) relationship. These cause problems related to relationship.

For example, for $Q_{4.2} = \{\text{PARTA}, 100\}$, the subtree rooted at Part_(46) is an answer. However, this is not complete since price 100 is the price of a part named PARTA supplied by Supplier_(41). It is not the price of Part_(46). Thus, the answer should be moved up to Supplier_(41) to include Supplier_(41) as well.

For another example related to ternary relationship, $Q_{4.3} = \{\text{PARTA}, 150\}$, the answer is the subtree rooted at `Part_(36)`. This is not complete either since 150 is the quantity of a part named PARTA supplied by `Supplier_(41)` to `Project_(21)`. Quantity is not an attribute of object `Part_(46)`. Thus, the answer should be moved up to `Project_(21)` to include `Project_(21)` and `Supplier_(41)`.

4.2 Problems Can Be Solved with IDREF

IDREF mechanism is based on semantics of object and object ID, thus using IDREF can avoid problems caused by lack of semantics of object, including the problems of *missing answer*, *duplicated answer* and *schema dependence*. However, if ID Reference mechanism is not totally applied for all objects, i.e., there exists some objects without IDREF as object `<Employee : HT08>` in Fig. 3, then the above problems are not totally solved.

For example, $Q_{4.4} = \{\text{Bill}, \text{HT08}\}$ has two duplicated answers, `Employee_(6)` and `Employee_(26)`. For $Q_{4.5} = \{\text{Prj2012}, \text{Prj2013}\}$, only the subtree containing `Supplier_(41)` can be returned whereas the subtree containing `<Employee : HT08>` is *missed*. If object class `Employee` is designed as the parent of object class `Project`, the missing answer of $Q_{4.5}$ are found. It shows that the graph-based search also *depends on the design of XML schema* in this case.

Summary. The graph-based search can avoid *missing answer*, *duplicated answer* and *schema dependence* only if the ID reference completely covers all objects. Otherwise, the above limitations cannot avoid. The other problems including *meaningless answer* and *problems related to relationship* are still unsolved no matter IDREFs are used or not because IDREF mechanism only considers semantics of object and object ID but ignores semantics of relationship and attribute.

5 Impact of ORA-Semantics in XML Keyword Search

We pointed out limitations of the structure-based search (the LCA-based and the graph-based approaches) because of unawareness of identification of object, relationship and attribute. We refer such identification as ORA-semantics. This section introduces ORA-semantics, shows the impact of ORA-semantics in XML keyword search and discusses the way to discover ORA-semantics from XML schema and data.

5.1 ORA-Semantics

The term *semantics* has different interpretations. In this paper, we define the concept of ORA-semantics to include the identification related to *object*, *relationship* and *attribute*. At data level, an *object* represents a real world entity. Several objects may be connected through some *relationship*. Objects and relationships may have a set of *attribute values* to describe their properties. Object, relationship and attribute value is an instance of *object class*, *relationship type* and *attribute* respectively at the schema level. Besides such major semantics, there are connecting nodes such as composite attributes or aggregation nodes. In brief, ORA-semantics is defined as follows.

Concept 6 (ORA-semantics (Object-Relationship-Attribute-semantics)). In an XML schema tree, the ORA-semantics is the identification of object class, OID, object attribute, aggregation node, composite attribute and explicit/implicit relationship type with relationship attributes.

For example, at schema level, ORA-semantics of the schema in Fig. 1 includes:

- Student, Course and Textbook are object class
- Student_No, Code and ISBN are object ID of the above object classes.
- Grade is the attribute of the relationship between Student and Course.
- For simplicity, we do not include object attribute in the schema in Fig. 1. The hidden object attributes include Student_No, Name, etc of object class Student.

At data level, ORA-semantics in the XML data in Fig. 1 includes that Course_(11) is object; CS5201 is its object ID value; Database, Computing are its attribute values; especially A is an attribute value of the relationship it involves in, etc.

5.2 Impact of ORA-Semantics in XML Keyword Search

Object semantics. Object identification helps detect multiple occurrences of the same object (duplicated object) appearing at different places in XML document. This enables us to filter *duplicated answers* and discover *missing answers*.

Relationship semantics. Relationship identification helps discover the degree of a relationship to return a more complete answer for queries involving in *ternary relationship*.

Attribute semantics. Differentiating object attribute and relationship attribute avoids returning incorrect answer for queries involving in *relationship attribute*. Moreover, differentiating object and non-object node also avoids *meaningless answers*.

ORA-semantics. Exploiting ORA-semantics in processing keyword query provides answers independent from the schema designs. In brief, ORA-semantics can resolve all problems of the structure-based search discussed in Sec. 3 and Sec. 4.

5.3 Discovering ORA-Semantics

ORA-semantics includes the identification of internal nodes and leaf nodes in XML schema. Particularly, an internal node can be classified into object class, explicit relationship type, composite attribute and aggregation node. A leaf node can be a relationship attribute or an object attribute, which further can be object ID or a normal object attribute. We have designed algorithms to automatically discover ORA-semantics with high accuracy for overall process (higher than 94%). More details can be found in [14]. Other effective algorithms can be studied; however, this task is orthogonal to this paper.

6 Our Semantics-Based XML Keyword Search

Discussion in Sec. 4.1 shows that even if all objects follow IDREF mechanism and thus there is no duplication, the existing graph-based search still suffers from problems of

meaningless answer, and especially *problems related to relationship*. In brief, IDREF cannot solve all problems of the existing structure-based search. To solve these problems, a more semantics-enriched model is needed, in which not only objects, but relationships and their attributes must be fully captured as well. We illustrate this model by proposing Object Relationship (OR) graph to represent an XML document. Both objects and relationships are represented as nodes in an OR graph while attributes and values are associated with the corresponding objects and relationships. As such, an OR graph can capture all objects, relationships and attributes of an XML data.

In this section, we first introduce the OR graph, its advantages, and the process of OR graph generation. We then present search semantics, i.e., formally define the expected answers, and query processing based on the OR graph. To show the advantages of the OR graph, we compare our OR graph based search with the structure-based search. To improve efficiency, we propose indexes and optimized search algorithms.

6.1 Object Relationship (OR) Graph

Definition 1 (OR graph). An OR graph $G = (V_O, V_R, E)$ is an unweighted, undirected bipartite graph with two types of nodes, where V_O is the set of object nodes, V_R is the set of relationship nodes, $V_O \cap V_R = \emptyset$, and E is the set of edges. An edge is between an object node in V_O and a relationship node in V_R . For every relationship $r \in V_R$, there is an edge between each of its participating objects and r .

The information stored for each object node is a quadruple $\langle \text{nodeID}, \text{object class}, OID, \text{associated keywords} \rangle$. The information stored for each relationship node is a similar quadruple $\langle \text{nodeID}, \text{relationship type}, \{\text{participating } o_i\}, \text{associated keywords} \rangle$. Each object/relationship has a randomly generated *nodeID* as label in the OR graph. Associated keywords of an object/relationship include its attributes and values as well.

Consider the XML data in Fig. 3. The OR Graph conforming to this XML data is shown in Fig.4, where *square* and *diamond* stand for *object node* and *relationship node* respectively. Each object/relationship node in an OR graph has an identifier, e.g., o_1, o_2, o_3, o_4 and o_5 are objects and r_1, r_2, r_3, r_4 and r_5 are relationships. Additionally, to serve for a clearer explanation, for each object node in Fig.4, we associate its object class and object ID (e.g., Project Prj1 for o_1). Moreover, we also show matching nodes of query keywords discussed in Sec. 4 (e.g., Amazon matches o_4).

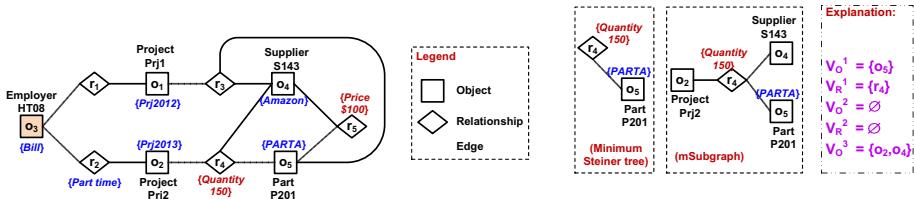


Fig.4. The OR graph w.r.t. the XML data in Fig. 3

Fig.5. *mSubgraph* for $Q_{4.3}$

6.2 Features and Advantages of the OR Graph

FA1. Object-relationship level. Our OR graph represents XML document at object-relationship level by associating attributes and values with their corresponding objects and relationships. This can significantly *reduces the search space* since the number of nodes of an OR graph is much less than that of the corresponding XML data due to not counting attributes, values and duplicated objects. Moreover, it can avoid *meaningless answers* because an answer must correspond to a whole object or relationship rather than an arbitrary XML element.

FA2. Duplicate-free. An object may appear as multiple occurrences in an XML document. In an OR graph, each object node represents an *object*, not an occurrence of object. Thus, an object is not duplicated because it corresponds to only one object node. Duplicate-free is a very important feature of OR graph, by which the process can find *missing answers* and avoid *duplicated answers* from the LCA-based approach.

FA3. Explicit appearance of relationships. Differentiating object nodes and relationship nodes enables us to distinguish between object attribute and relationship attribute. This avoids returning incorrect answers for queries involving relationship attribute. Moreover, relationship nodes can provide the degree (e.g., binary or ternary) of a relationship. As a result, we can add *all participating object nodes* of relationship nodes in a returned answer to make it more meaningful.

FA4. Schema-independence. The OR graphs conforming to any XML representations of the same data source are the same no matter which schema design is used. In other words, the OR graph is *independent* of the XML structure. This enables us to return the same answers for all designs of the same data source. For example, no matter <Paper : ER32> in the data in Fig. 3 is designed with or without IDREF, the OR graphs conforming is in Fig. 4.

6.3 Generating OR Graph from an XML Document

Algorithm 1. OR graph generation

```

Input:  $D$ : XML document
Output: OR graph  $G(V_O, V_R, E)$ 
1  $V_O \leftarrow \emptyset$ 
2  $V_R \leftarrow \emptyset$ 
3  $E \leftarrow \emptyset$ 
4 for each object  $o$  visited by document order
do
5   if  $o$  is not duplicated with objects in
       $V_O$  then
6      $V_O.Add(o)$ 
7    $Rel(o) \leftarrow$  relationships among  $o$  and
      its parents
8   for each relationship  $r$  in  $Rel(o)$  do
9     if  $r$  is not duplicated with
        relationships in  $V_R$  then
10     $V_R.Add(r)$ 
11     $e \leftarrow$  edge between  $o$  and  $r$ 
12     $E.Add(e)$ 

```

To generate OR graph $G(V_O, V_R, E)$ from an XML document D (with or without IDREFs), we identify V_O , V_R , E from nodes in D . Only distinct objects in D are added to V_O . The process of OR graph generation is presented in Algorithm 1. We traverse XML document by document order. For each visited object node o , we add o to V_O if o is not duplicated with any existing object in V_O and find the the set of relationships $Rel(o)$ among o and its parents. For each relationship r in $Rel(o)$, we add r to V_R if r is new. We finally add the edge between o and r to E . In brief, the sequence is adding an object, then a relationship it participates in, and finally the edge between them [11].

6.4 Search Semantics

This paper adopts the *minimum Steiner tree semantics* [5,6] because it is the widely accepted semantics for the graph-based search. A Steiner tree is a subtree of the data graph that contains all keywords. The weight of a Steiner tree is defined as the total weight of its edges. A *minimum* Steiner tree is a Steiner tree having the smallest weight among all the Steiner trees w.r.t. the same set of matching nodes. However, a subtree returned by this semantics may not contain completely meaningful information to answer a query, especially when XML data has complicated structure. Therefore, we determine what other nodes should be added in order to return a more meaningful answer to users. Consequently, we extend a minimum Steiner tree s to become a more meaningful subgraph (called $mSubgraph$) by *adding all participating objects* of all relationships in s . The rationale is that a relationship itself has no or incomplete meaning without all of its participating objects. An answer $mSubgraph$, is defined as follows.

Definition 2 (mSubgraph). Given a keyword query Q to the OR graph $G = (V_O, V_R, E)$. An answer $mSubgraph$ of Q is a subgraph of G and is denoted as $mS(mV_O, mV_R, mE)$, where $mV_O = V_O^1 \cup V_O^2 \cup V_O^3$ and $mV_R = V_R^1 \cup V_R^2$ such that

- V_O^1 and V_R^1 are the sets of matching objects and relationships.
- V_O^2 and V_R^2 are the sets of intermediate objects and relationships connecting objects in V_O^1 and relationships in V_R^1 .
- V_O^3 is the set of added objects which participate in mV_R but are not in $V_O^1 \cup V_O^2$.
- mE is the set of edges between each object in mV_O to its relationships in mV_R .

Let us recall $Q_{4.3} = \{\text{PARTA, 150}\}$ to illustrate the benefits of $mSubgraph$. Answers for this query (both in form of a minimum Steiner tree and in form of an $mSubgraph$) are shown in Fig. 5. As can be seen, $mSubgraphs$ are more meaningful than minimum Steiner trees with intuitive meaning of 150 parts name PARTA are supplied by <Supplier:S143> to <Project:Prj1> while the minimum Steiner tree does not provide information about the supplier and the project.

6.5 Comparison on the LCA, Graph and Semantics Based Approaches

We compare the limitations of the LCA-based, the graph-based and our OR graph based search and show the reasons behind in Table 1.

6.6 Query Processing

This work aims to show the impact of ORA-semantics on effectiveness of XML keyword search. The ORA-semantics is exploited in generating the OR graph. After the generation process, searching over OR graph to find minimum Steiner trees can be implemented by existing efficient algorithms. Answers then will be extended to $mSubgraphs$. Space limitation precludes detailed discussion about their algorithms. However, readers may find them in [5,6]. We also design our own algorithm and propose index and an optimized techniques to improve the efficiency in Sec. 6.7. We work at object level by assigning nodes describing an object instance the same label. Thereby, we dramatically reduce the search space and greatly improve the efficiency.

Table 1. Comparison on the LCA-based, graph-based and OR graph based approaches

Problem	LCA-based	Graph-based	Reason of the problems of the structure-based search	Semantics needed	OR graph based	Reason that the semantic based search can avoid problems
Meaningless answer	Yes	Yes	Returning non-object element because of not differentiating object and non-object node	-Object -Attribute	No	All attributes and values are associated with objects/relationships (Object-relationship level, FA1)
Missing answer	Yes	Partial	- Unable to discover the same object appearing at different places in XML document	Object	No	The same objects can be discovered (Duplicate-free, FA2 of OR graph)
Duplicated answer	Yes	Partial	- Partial because it can be solved with IDREF			
Problems related to relationships	Yes	Yes	- Unable to distinguish relationship attribute and object attribute - Not aware ternary relationships and above	-Attribute -Relationship	No	Relationships and relationship attributes are discovered (Explicit appearance of relationships, FA3 of OR graph)
Schema dependence	Yes	Partial	- Relying on hierarchy (LCA) - Partial because it can be solved with IDREF	Object	No	Different designs of the same data have the same ORA-semantics (Schema-independence, FA4)

6.7 Index and Optimization

Indexes. To make distinction between query keywords with document keywords, we call the latter as *term*. To support our optimized algorithm, we propose three indexes: semantic-inverted lists, term-node lists and node-node lists. Due to space constraint, we very briefly describe these indexes. Details are given in our technical report [11].

Semantic-inverted list is to store the set of matching objects and the set of matching relationships of a term. Term-node lists is to store the shortest distance from some node to $mNode(t)$ where $mNode(t)$ is the set of nodes matching term t . Node-node list is to store a set of shortest distances from some node to each node in $mNode(t)$.

The optimized algorithm. The search has two phases. To find a good Steiner tree, we first find a good intermediate node v in sense that it can connect to all keywords with short distances. From v , we then generate the Steiner tree by look up the closest matching node for each keyword. The three indexes enable us to propose very efficient algorithms for the two-phase search. Details of explanations are given in [11].

Algorithm 2. Optimized Algo.

```

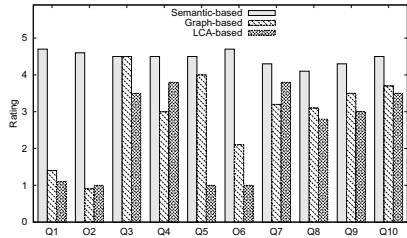
Input: Query  $Q = \{k_1, k_2, \dots, k_n\}$ 
Term-node lists  $L_{TN}$ 
Output: top-k answers in  $Ans(Q)$ 
1 Variables:  $T_{prune}$ : pruning threshold
2  $T_{prune} \leftarrow 0$ 
3 for  $i \in [1, n]$  do
4    $c_i \leftarrow$  new Cursor( $L_{TN}(k_i), 0$ )
5 while
6    $\exists j \in [1, n] : c_j.Next() \neq NULL$  do
7      $i \leftarrow$  pick from  $[1, n]$  by BF order;
8      $< v > \leftarrow c_i.Next()$ 
9     if  $v \neq NULL$  then
10       KeywordNavigation( $v$ );
11       if  $c_i.Dist() > T_{prune}$  and
12          $T_{prune} \geq 0$  then
13           Output top-k answers in
14              $Ans(Q)$ . Exit
15            $T_{prune} \leftarrow$  the  $k^{th}$  biggest of
16             {size( $\alpha$ ) |  $\alpha \in Ans(Q)$ }
```

Algorithm 3. KeywordNavigation(v)

```

Input: a connecting node  $v$ 
Node-node lists  $L_{NN}$ 
1 Variables:  $cNode(Q)$ : connecting nodes visited, initially  $\emptyset$ 
2 if  $cNode(Q).contain(v)$  then
3   return
4    $cNode(Q).add(v)$ 
5 while  $TRUE$  do
6   for  $j \in [1..n]$  do
7      $< u_j > \leftarrow$  pick from  $L_{NN}(v)$  in BestFS order;
8      $g \leftarrow \langle \{u_1, u_2, \dots, u_n, \}, V_I, E \rangle$ 
9     if  $g \notin Ans(Q)$  then
10       if size( $g$ )  $> T_{prune}$  and  $T_{prune} \geq 0$  then
11          $T_{prune} \leftarrow$  the  $k^{th}$  biggest of
12           {size( $\alpha$ ) |  $\alpha \in Ans(Q)$ }
```

Query	Query Keywords	Dataset
Q1	wizbang4	eBay
Q2	Bill	NBA
Q3	id_num, 511364992	eBay
Q4	ct-inc, CyberTech	eBay
Q5	Michael, Coach	NBA
Q6	Player, Bill, Sam	NBA
Q7	pizarju01, teams	Baseball
Q8	BOS, TOR, vioxji01	Baseball
Q9	Celtics, player	NBA
Q10	Michael, Celtics, team	NBA

Fig. 6. 10 keyword queries for users to rate**Fig. 7.** Scores of answers

7 Experiments

We compare the quality of answers returned by our OR graph based search with XK-Search [22] and BLINK [7] (two state-of-the-art algorithms to represent the LCA-based and graph-based search respectively). Using these two algorithms already show the improvement of our semantic-based search over the structure-based search because other structure-based approaches suffer from the same problems with them. The experiments were performed on a Intel(R) Core(TM)2 Duo CPU 2.33GHz with 3.25GB of RAM with three real data sets: eBay¹ (0.36MB), NBA²(45.2MB), Baseball³ (60MB).

7.1 Rating Answers

We compare the quality of the answers returned by our approaches and the structure-based approaches by rating their answers. We randomly generated 25 queries from all document keywords. After filtering out some meaningless queries, we chose 10 queries as shown in Table. 6. We asked 21 students major in computer science to rate the top-10 answers on scale [0-5] (0 for totally mismatch and 5 for perfectly match the user expectation). The scores of answers are shown in Fig. 7.

Discussion. As shown, our approach gets the highest scores for all queries, which means that our approach returns more meaningful answers to users. Moreover, the scores of our approach are very high (above 4) and stable, which infers that users are satisfied with our answers. In contrast, the structure-based search gets lower scores since its answers mismatch the expectations of users. Especially for queries Q_1 , Q_2 and Q_6 , the scores are around 1 because they returns hundreds of duplicated attribute nodes without any detailed information. Generally, the graph-based search has higher score than the LCA-based search because it can find missing answers and avoid duplicated answers when IDREFs are considered.

7.2 Statistics on Problems of Answers

We collected 86 keyword queries for the above three data sets: 12 queries for eBay, 44 queries for NBA and 30 queries for Baseball from 21 students working in computer

¹ www.cs.washington.edu/research/xmldatasets/data/auctions/ebay.xml

² <http://www.databasebasketball.com/>

³ <http://www.seanlahman.com/baseball-archive/statistics/>

science. Based on the discussion in Sec. 3 and Sec. 4, the answers which cannot match the users' search intention are classified into five categories: including (A) *Meaningless answer*; (B) *Missing answer*; (C) *Duplicate answer*; (D) *Relationship problem*; and (E) *Schema dependence*. Fig. 8 shows the percentage of answers containing a certain problem over total answers. Note that an answer may contain more than one problem.

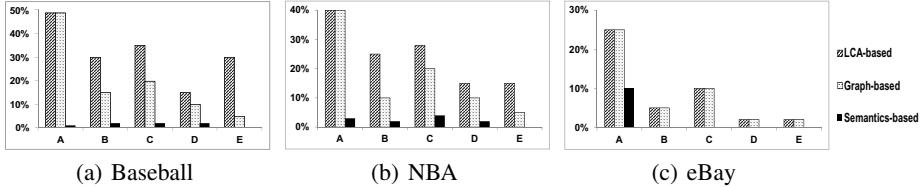


Fig. 8. Statistics on problems of answers: (A) *Meaningless answer*; (B) *Missing answer*; (C) *Duplicate answer*; (D) *Relationship problem*; (E) *Schema dependence*

Discussion. Generally, answers of our approach have fewer problems than those of the structure-based search. The most frequent problem is the meaningless answer. It usually occurs when a query contains only one keyword and the structure-based search returns only one non-object node matching that keyword. Sometimes our approach returns a meaningless answer when it discovers a composite attribute as an object. Missing and duplicated answers are also frequent problems because the structure-based search cannot discover the same object appearing in multiple nodes.

7.3 Efficiency

We evaluate the efficiency of the three compared works by varying the number of query keywords and node frequency of keyword using NBA dataset. The response time is shown in Fig. 9.

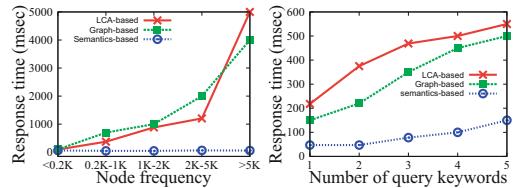


Fig. 9. Response time

Discussion. The response time of our approach depends on the frequency of matching objects and relationships rather than the node frequency since it works at object and relationship level. Thus, our approach runs stably while the others' response time increases very fast with the node frequency. Moreover, working at object and relationship level largely reduces the search space and thus enables our approach to run much faster than the others. More explanations and experiments on efficiency are given in [11].

8 Related Work

Tree-based XML keyword search. Most existing tree-based XML keyword search methods are LCA-based and depend on hierarchical structure of the data. XKSearch [22] defines Smallest LCAs (SLCAs) to be the LCAs that do not contain other LCAs.

Meaningful LCA (MLCA) [15] incorporates SLCA into XQuery. VLCA and ELCA [12,23] introduce the concept of valuable/ exclusive LCA to improve the effectiveness of SLCA. MESSIAH [20] handles cases where there are missing values in optional attributes. Although researchers have put efforts on improving LCA-based effectiveness, their works are still based on the hierarchical structure without ORA-semantics. Thus, they face problems as studied in Sec. 3.

Graph-Based XML Keyword Search. Minimum Steiner tree and distinct root semantics [3,7,9] are popular but may return answers whose content nodes are not closely related. Recently, subgraph semantics [13,19] and content based semantics [10,16] is proposed to handle the above problem, but they still do not consider ORA-semantics. Thus, they face problems as discussed in Sec. 4.

Semantics-Based XML Keyword Search. XSEarch [4] focuses on adding semantics into query but the added semantics is for distinguishing a tag name and a value keyword only. XSeek [17] and MaxMatch [18] infer semantics from keyword query. They can only infer semantics of object since it is impossible to infer any semantics of object ID, relationship and relationship attribute from a keyword query. XKeyword [8] exploits semantics from the XML schema. XReal [1], Bao et. al. [2] and Wu et. al. [21] proposed an object-level for XML keyword search. However, all of these works only consider objects, but they do not have the concepts of object ID, relationship and attribute. Therefore, they can avoid at most the problem of meaningless answer but still suffer from all other problems discussed in Sec. 3 and 4.

9 Conclusion and Future Work

We have systematically illustrated limitations of the existing LCA-based search, including the problems of *meaningless answer*, *missing answer*, *duplicate answer*, *problems related to relationship*, and *schema dependence* which are caused by unawareness of semantics of object, relationship and attribute. We have also demonstrated that even with IDREFs, the graph-based search can avoid at most the problems of *missing answer*, *duplicate answer* and *schema dependence* because IDREF mechanism is aware of only semantics of object and object ID but not semantics of relationship and attribute. Thus, the graph-based search still suffers from the problems of *meaningless answer* and *problems related to relationship*. We introduced *ORA-semantics* which is semantics of object, relationship and attribute and showed its importance in XML keyword search. To take ORA-semantics into account for XML keyword search, we propose Object Relationship (OR) graph to represent XML data, in which objects and relationships correspond to nodes, while attributes and values are associated with the corresponding object/relationship nodes. As such, OR graph can capture all ORA-semantics of an XML data. To process a query, we search over the OR graph. Our index and optimization provide an efficient search algorithm. Experimental results showed that our OR graph based approach outperforms the structure-based search in term of both effectiveness and efficiency. Thus, semantics-based approach could be a promising direction for XML keyword search in solving problems of the current structure-based search.

References

1. Bao, Z., Ling, T.W., Chen, B., Lu, J.: Efficient XML keyword search with relevance oriented ranking. In: ICDE (2009)
2. Bao, Z., Lu, J., Ling, T.W., Xu, L., Wu, H.: An effective object-level XML keyword search. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5981, pp. 93–109. Springer, Heidelberg (2010)
3. Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., Sudarshan, S.: Keyword searching and browsing in databases using BANKS. In: ICDE (2002)
4. Cohen, S., Mamou, J., Kanza, Y., Sagiv, Y.: XSEarch: A semantic search engine for XML. In: VLDB (2003)
5. Ding, B., Yu, J.X., Wang, S., Qin, L., Zhang, X., Lin, X.: Finding top-k min-cost connected trees in database. In: ICDE (2007)
6. Golenberg, K., Kimelfeld, B., Sagiv, Y.: Keyword proximity search in complex data graphs. In: SIGMOD (2008)
7. He, H., Wang, H., Yang, J., Yu, P.S.: BLINKS: ranked keyword searches on graphs. In: SIGMOD (2007)
8. Hristidis, V., Papakonstantinou, Y., Balmin, A.: Keyword proximity search on XML graphs. In: ICDE (2003)
9. Kacholia, V., Pandit, S., Chakrabarti, S., Sudarshan, S., Hrishikesh Karambelkar, R.D.: Bidirectional expansion for keyword search on graph databases. In: VLDB (2005)
10. Kargar, M., An, A.: Keyword search in graphs: finding r-cliques. In: PVLDB (2011)
11. Le, T.N., Wu, H., Ling, T.W., Li, L., Lu, J.: From structure-based to semantics-based: Effective XML keyword search. TRB4/13, School of Computing, NUS (2013)
12. Li, G., Feng, J., Wang, J., Zhou, L.: Effective keyword search for valuable LCAs over xml documents. In: CIKM, pp. 31–40 (2007)
13. Li, G., Ooi, B.C., Feng, J., Wang, J., Zhou, L.: EASE: Efficient and adaptive keyword search on unstructured, semi-structured and structured data. In: SIGMOD (2008)
14. Li, L., Le, T.N., Wu, H., Ling, T.W., Bressan, S.: Discovering semantics from data-centric XML. In: Decker, H., Lhotská, L., Link, S., Basl, J., Tjoa, A.M. (eds.) DEXA 2013, Part I. LNCS, vol. 8055, pp. 88–102. Springer, Heidelberg (2013)
15. Li, Y., Yu, C., Jagadish, H.V.: Schema-free XQuery. In: VLDB (2004)
16. Liu, X., Wan, C., Chen, L.: Returning clustered results for keyword search on xml documents. In: TKDE (2011)
17. Liu, Z., Chen, Y.: Identifying meaningful return information for XML keyword search. In: SIGMOD (2007)
18. Liu, Z., Chen, Y.: Reasoning and identifying relevant matches for XML keyword search. In: PVLDB (2008)
19. Qin, L., Yu, J.X., Chang, L., Tao, Y.: Querying communities in relational databases. In: ICDE (2009)
20. Truong, B.Q., Bhowmick, S.S., Dyreson, C.E., Sun, A.: MESSIAH: missing element-conscious slca nodes search in xml data. In: SIGMOD (2013)
21. Wu, H., Bao, Z.: Object-oriented XML keyword search. In: Jeusfeld, M., Delambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 402–410. Springer, Heidelberg (2011)
22. Xu, Y., Papakonstantinou, Y.: Efficient keyword search for smallest LCAs in XML databases. In: SIGMOD (2005)
23. Zhou, R., Liu, C., Li, J.: Fast ELCA computation for keyword queries on XML data. In: EDBT (2010)

Combining Personalization and Groupization to Enhance Web Search

Kenneth Wai-Ting Leung¹, Dik Lun Lee¹, and Yuchen Liu²

¹ Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Hong Kong
`{kwtleung, dlee}@cse.ust.hk`

² Computer Science Department, University of California, Los Angeles
`yliu@cs.ucla.edu`

Abstract. To improve retrieval effectiveness, personalized search engines adjust the search results according to the user’s interest profile. Most of the existing work are based on either pure personalization or pure groupization. Personalization tries to produce results aligned with the user’s interests, whereas groupization aims to broaden the results using the interests of the user’s communities. In this paper, we propose to combine personalization and groupization to improve the retrieval effectiveness of a personalized search engine. We observe that recommendations derived from a user’s communities may be too broad to be relevant to a user’s specific query. Thus, we study different ways to refine user communities according to the user’s personal preferences to improve the relevance of the recommendations. We introduce online user community refinement to identify highly related users and use their preferences to train a community-based personalized ranking function to improve the search results. To produce the user communities, we propose the Community Clickthrough Model (*CCM*) to model search engine clickthroughs as a tripartite graph involving users, queries and concepts embodied in the clicked pages, and develop the Community-based Agglomerative-Divisive Clustering (*CADC*) algorithm for clustering the *CCM* graph into groups of similar users, queries and concepts to support community-based personalization. Experimental results show that a refined user community that only uses *focused* users’ recommendations can significantly improve *nDCG* by 54% comparing to the baseline method. We also confirm that *CADC* can efficiently cluster *CCM* to enhance a personalized search engine.

1 Introduction

As the web grows in size, it has become increasingly difficult to find highly relevant information to satisfy a user’s specific information needs. Given a query, most commercial search engines return roughly the same results to all users. However, different users may have different information needs even for the same query. For example, a user who is looking for a mobile phone may issue a query “apple” to find products from Apple Inc., while a housewife may use the same query “apple” to find apple recipes. To improve retrieval effectiveness, personalized search engines create user profiles on the users’ preferences, which are used to adjust the search results to suit the users’ preferences.

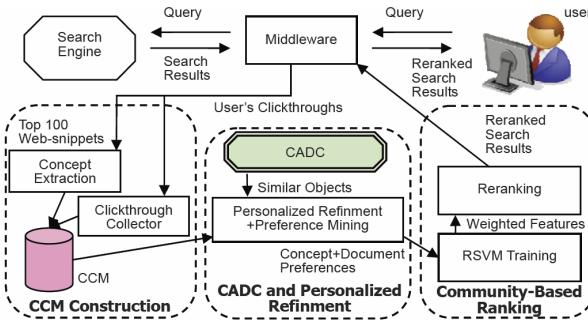


Fig. 1. Flow of the CADC community-based personalization

Most personalized search engines analyze clickthrough data to extract the users' preferences. A challenge of personalization lies in collecting clickthrough data rich enough to understand the precise user preferences. In groupization, users of similar interests form user communities and the collective clickthroughs of a group are used to augment a user's interest profile, and thus enhance the accuracy of search by adapting the ranking function to rank relevant results higher in the result list [13]. The objective of groupization is to disambiguate a query (e.g., “apple” in the above example) according to the user's group and to return relevant results with respect to the query and the user group.

We observe that most of the existing work perform either pure personalization or pure groupization. Pure personalization adjusts results presented to the user based on her search histories and hence tends to recommend or promote results that the user is already familiar with. Pure groupization adjusts the results based on the interests of a community but suffers from irrelevant results that are far from the user's current interest because a community always exhibits differences in interest and behavior among its members. To address the problems of personalization and groupization, we propose to study different ways to refine user communities according to a user's personal preferences to ensure the relevance of the recommendations. We aim to strike a good balance between personalization and groupization in order to obtain augmented user profiles that are highly relevant to users' personal preferences. The community-based collaborative search engine proposed in this paper consists of several important processes. Figure 1 shows the overall flow between the processes.

We study different refined user communities and conduct extensive experiments to compare the quality of their recommendations. Experimental results confirm our community-based personalization method using the refined user community with highly similar *focused* users only improves retrieval effectiveness significantly comparing to the baseline method.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we present our community clickthrough model (CCM) model search engine clickthroughs as a tripartite graph involving users, queries and concepts embodied in the clicked pages. In Section 4, we describe the CADC algorithm for clustering the CCM graph into groups of similar users, queries and concepts to support community-based personalization.. In Section 5, we identify refined user communities from the clustered CCM. In Section 6, we present C-RSVM for learning a community-based personalized

ranking function. Experimental results evaluating the performance of CADC against state-of-the-art methods are presented in Section 7. Section 8 concludes the paper.

2 Related Work

In this section, we first review a few state-of-the-art techniques for clustering of clickthrough data in Section 2.1. Then, we describe the applications of the user, query, and document clusters in Section 2.2.

2.1 Clickthrough Clustering

Beeferman and Burges [2] proposed an agglomerative clustering algorithm, which models clickthrough data as a query-document bipartite graph with one set of nodes corresponding to the set of the submitted queries. It iteratively merges similar objects (i.e., queries or documents) together until the termination condition is satisfied. Later, Sun, et. al. proposed CubeSVD [12] to model the relationships between users, queries, and documents by representing the clickthrough data as a 3-order tensor. A Higher-Order Singular Value Decomposition (*HOSVD*) technique is employed to discover latent relationships among objects in the 3-order tensor.

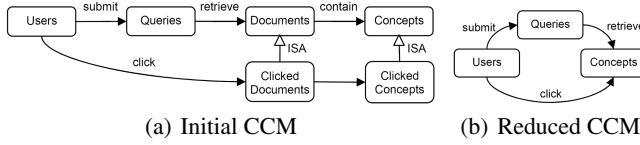
Wang, et. al. [15] proposed M-LSA, which represents the relationships between users u , queries q , and documents d with three co-occurrence matrices ($M_{u \times q}$, $M_{u \times d}$, and $M_{q \times d}$). M-LSA employs Eigen Value Decomposition (*EVD*) to discover important objects from the matrices.

2.2 Applications of the Resulting Clusters

Clustering of clickthrough data generates groups of similar users, similar queries, and similar documents. Similar users can be used to form user communities for collaborative filtering to predict the interests of a user. Teevan, et. al. [13] conducted a study in community-based personalization that combines clickthroughs from related users to enhance the performance of personalized search. Our work is similar to their work; however, Teevan, et. al. did not provide an automatic mechanism to discover user preferences from the clickthrough data. The user groups are manually generated according to the explicit information provided by the users or implicit information inferred from users' activities. Similar queries can be suggested to users to help them formulate more effective queries. Leung et al. [10] proposed an effective approach to generate personalized query suggestions. Finally, the similar documents can be used to categorize the retrieved documents. In Ferragina and Gulli's work [3], a hierarchical clustering engine (*SnakeT*) was proposed to organize search results into a hierarchy of labeled folders.

3 Community Clickthrough Model

The three clustering methods (BB, CubeSVD, and M-LSA) discussed in Section 2 are content-ignorant in that two queries are related if they induce clicks on the same document, but completely ignore the content of the documents. One major problem with content-ignorant model is that the number of common clicks on documents induced

**Fig. 2.** Initial CCM and our Reduced CCM**Table 1.** Example Concepts Extracted for the Query “apple”

c_i	$support(c_i)$	c_i	$support(c_i)$	c_i	$support(c_i)$	c_i	$support(c_i)$
iPod	0.3	macintosh	0.1	iTunes	0.06	apple orchard	0.03
iPhone	0.2	apple store	0.07	fruit	0.04	apple farm	0.02

by different queries is very small. Beeferman and Berger [2] reported that the chance for two random queries to have a common click is merely 6.38×10^{-5} [2] in a large clickthrough data set from a commercial search engine. Thus, the bipartite graph or the co-occurrence matrix would be too sparse for obtaining useful clustering results.

To alleviate the click sparsity problem, we introduce a content-aware clickthrough model, called Community Clickthrough Model (CCM). As illustrated in Figure 2(a), the initial CCM represents users submitting queries to the search engine and queries retrieving documents from the search engine. Users click on some of the retrieved documents based on the perceived relevance of the document snippets to their queries. Concepts are derived from the retrieved and clicked documents, transforming the model from document-oriented to concept-oriented. Finally, we remove the document layer in the initial CCM to obtain the Reduced CCM (see Figure 2(b)), which is a tripartite graph relating *users*, their submitted *queries*, the *retrieved concepts* and the *clicked concepts*, which are a subset of the retrieved concepts. Hereafter, CCM refers to *reduced CCM* unless stated otherwise. For clarity, when a user u_i clicks on a document that embodies a concept c_l , we simply say u_i *clicks* on c_l ; if q_j retrieves a document that embodies concept c_l , we say q_j *retrieves* c_l .

It is clear that the use of concepts instead of documents can resolve the click sparseness problem [10] because the chance that two different queries/users induce similar concepts and as such are considered similar is much higher than in a content-ignorant model. We will also show later that CCM resolves ambiguous queries due to synonyms and polysemy by distinguishing queries issued from users in different communities and hence improves precision.

To identify concepts embodied in a document, we define a concept as a sequence of one or more words that occur frequently in the web-snippets¹ of a particular query q using the support, defined as: $support(c_i) = (sf(c_i)/n) \cdot |c_i|$, where c_i is a particular keyword/phrase extracted from the web-snippets, $sf(c_i)$ is the snippet frequency of the keyword/phrase c_i (i.e., the number of web-snippets containing c_i), n is the number of web-snippets returned and $|c_i|$ is the number of terms in the keyword/phrase c_i . A similar concept extraction approach has also been used in [8] and [9].

¹ “web-snippet” denotes the title, summary and URL of a Web page returned by search engines.

After the concepts are extracted, the following rules are introduced to compute the similarity between objects:

1. Two users are similar if they submit similar queries and click on similar concepts.
2. Two queries are similar if they are submitted by similar users and retrieve similar concept.
3. Two concepts are similar if they are clicked by similar users and are retrieved by similar queries.

Based on the three rules, we propose the following similarity functions to compute the similarity between pair of users, pair of queries, and pair of concepts.

$$\text{sim}(u_i, u_j) = \frac{Q_{u_i} \cdot Q_{u_j}}{\|Q_{u_i}\| \|Q_{u_j}\|} \alpha_u + \frac{C_{u_i} \cdot C_{u_j}}{\|C_{u_i}\| \|C_{u_j}\|} (1 - \alpha_u) \quad (1)$$

$$\text{sim}(q_i, q_j) = \frac{U_{q_i} \cdot U_{q_j}}{\|U_{q_i}\| \|U_{q_j}\|} \alpha_q + \frac{C_{q_i} \cdot C_{q_j}}{\|C_{q_i}\| \|C_{q_j}\|} (1 - \alpha_q) \quad (2)$$

$$\text{sim}(c_i, c_j) = \frac{U_{c_i} \cdot U_{c_j}}{\|U_{c_i}\| \|U_{c_j}\|} \alpha_c + \frac{Q_{c_i} \cdot Q_{c_j}}{\|Q_{c_i}\| \|Q_{c_j}\|} (1 - \alpha_c) \quad (3)$$

where Q_{u_i} is a weight vector for the set of neighbor query nodes of the user node u_i in the CCM, the weight of a query neighbor node $q_{(k,u_i)}$ in the weight vector Q_{u_i} is the weight of the link connecting u_i and $q_{(k,q_i)}$ in the CCM. C_{u_i} is a weight vector for the set of neighbor concept nodes of the user node u_i in the CCM, and the weight of a query neighbor node $c_{(k,u_i)}$ in C_{u_i} is the weight of the link connecting u_i and $c_{(k,u_i)}$ in the CCM. Similarly, U_{q_i} is a weight vector for the set of neighbor user nodes of the query node q_i , C_{q_i} is a weight vector for the set of neighbor concept nodes of the query node q_i , U_{c_i} is a weight vector for the set of neighbor user nodes of the concept node c_i , and Q_{c_j} is a weight vector for the set of neighbor query nodes of the concept node c_i . The similarity of two nodes is 0 if they do not share any common node in the CCM, while the similarity is 1 if two nodes share exactly the same set of neighboring nodes.

CCM resolves the ambiguity problems such as synonyms and polysemy in the bipartite model. When u_1 submits the query “portable video” and clicks on the concept “ipod” and u_2 submits a (literally) different query “portable music” but clicks on the same concept “iPod”, they are considered as similar users even though their queries are literally different. This tackles the synonym problem. Further, u_3 and u_4 , who submit the same query “apple” but u_3 clicks on the concept “fruit” because he is interested in “apple” as a fruit and u_4 clicks on the concept “iPod” because he is interested in “apple” as a company, are considered having different interests because they click on different concepts even though their queries are literally the same. This resolves the polysemy problem.

4 CADC

Traditional agglomerative-only algorithms, which require the tripartite graph to be clustered all over again when new data are inserted into the graph, are not suitable for dynamic environments. Our Community-based Agglomerative-Divisive Clustering (*CADC*) algorithm performs updates efficiently by incrementally updating the tripartite graph as

new data arrives. It consists of two phases, namely, the **agglomerative phase** and **divisive phase**. The agglomerative phase iteratively merges similar clusters, while the divisive phase splits large clusters into small ones to prevent clusters from growing without bound when new data arrives.

4.1 Agglomerative Phase

The agglomerative phase is based on the three similarity assumptions as described in Section 3. The algorithm iteratively merges the two most similar users, then the two most similar queries, and then the two most similar concepts based on Equations (1), (2), and (3). The procedure repeats until no new cluster (user, query or document cluster) can be formed by merging.

4.2 Divisive Phase

The divisive phase employs a hierarchical clustering technique, which is an inverse of the agglomerative phase (splitting instead of merging). It iteratively splits large clusters into two smaller clusters until no new clusters can be formed by splitting. Basically, if a cluster contains very dissimilar objects, we will split the cluster into two smaller clusters that are more coherent. In the divisive phase, we adopt the splitting criterion as described in [11]. Assume that the top-most and second top-most dissimilar pairs of nodes in a cluster based on the distance Equations (4), (5), and (6), and denote them, respectively, as $d1_n = d(n_i, n_j)$ and $d2_n = d(n_k, n_l)$. Let $\Delta d = \frac{d(n_i, n_j) - d(n_k, n_l)}{d(n_i, n_j)}$, if Δd is larger than the split threshold δ , we will split the cluster into two smaller clusters using n_i and n_j as the pivots for the splitting.

Finally, to measure the distance between a pair of nodes in a cluster in order to compute ϵ_k , $d1_n = d(n_i, n_j)$, and $d2_n = d(n_k, n_l)$, we propose the following inverse of the similarity functions defined in Equations (1), (2), and (3), which are used as the distances between pair of users, pair of queries, and pair of concepts:

$$d(u_i, u_j) = \sqrt{\sum_{k=1}^n (q_{(k, u_i)} - q_{(k, u_j)})^2 \alpha_u} + \sqrt{\sum_{k=1}^m (c_{(k, u_i)} - c_{(k, u_j)})^2 (1 - \alpha_u)} \quad (4)$$

$$d(q_i, q_j) = \sqrt{\sum_{k=1}^n (u_{(k, q_i)} - u_{(k, q_j)})^2 \alpha_q} + \sqrt{\sum_{k=1}^m (c_{(k, q_i)} - c_{(k, q_j)})^2 (1 - \alpha_q)} \quad (5)$$

$$d(c_i, c_j) = \sqrt{\sum_{k=1}^n (u_{(k, c_i)} - u_{(k, c_j)})^2 \alpha_c} + \sqrt{\sum_{k=1}^m (q_{(k, c_i)} - q_{(k, c_j)})^2 (1 - \alpha_c)} \quad (6)$$

$q_{(k, u_i)} \in Q_{u_i}$ is the weight of the link connecting u_i and $q_{(k, u_i)}$, and $c_{(k, u_i)} \in C_{u_i}$ is the weight of the link connecting u_i and $c_{(k, u_i)}$. Similarly, $u_{(k, q_i)} \in U_{q_i}$, $c_{(k, q_i)} \in C_{q_i}$, $u_{(k, c_i)} \in U_{c_i}$, and $q_{(k, c_i)} \in Q_{c_i}$. Again, the combination thresholds α_u , α_q , and α_c are set to 0.5, in order to strike a good balance between the user, query, and concept dimensions.

4.3 The Clustering Algorithm

In CADC, the clickthrough data is first converted into a tripartite CCM G_3 and input to the algorithm. The algorithm iteratively merges and splits nodes in G_3 until the termination condition is reached and output the final clusters as G_3^C . When new clickthrough data arrives, new users, queries, and concepts are added to G_3^C as singleton

nodes and new links are added to G_3^C accordingly as G'_3 . G'_3 is then served as input to the clustering algorithm again. The new nodes are grouped to the correct clusters by the agglomerative phase. If a particular cluster becomes too large because too many new nodes are added to the cluster, the divisive phase will divide the cluster into smaller ones according to the statistical confidence given by Hoeffding bound. When the termination condition is reached, the algorithm outputs the updated tripartite CCM G_3^{CC} . Interested readers are referred to [6] for the detailed CADC algorithm.

5 Personalized Refinement

Up to this point, CADC clusters similar users into communities, which can be used to meet our objective of producing community-based ranking. However, user communities produced by CADC is based on the clickthrough data of the entire user population, which could be very large in a production environment and thus reflects the *general interest* of the users. The resulting user communities could be too large and too general to make high-quality recommendation for a particular query issued from a particular user (i.e., the *active user*). For example, given a user community interested in Apple products, a user issuing a query on “iPhone” may get recommendations from users interested in “iPad”, which may not be of interest to the user. While CADC can be used to further refine the community into more specific ones, it is impossible to have communities which are precise enough for every possible query and yet produce useful recommendations. In this section, we introduce online personalized refinement which dynamically extracts a subset from a user community based on (i) a user’s personal preferences and (ii) the clickthrough behaviors of the users to obtain high-quality recommendations.

5.1 Query-Based Refined User Communities

When a user u_i submits a query q_j , the following query-based refined user communities can be obtained by dividing the u_i ’s community based on q_j .

- $U_{(u_i, q_j)}$: Set of users who are similar to u_i and have submitted queries that are similar to q_j . They are called the **precise** users.
- U_{u_i} : Set of users similar to u_i (i.e., $U_{(u_i, q_j)} \cup U_{(u_i, \overline{q_j})}$). They are called the **suggestive** users.
- $U_{(u_i, \overline{q_j})}$: Set of users similar to u_i but have not submitted any query that is similar to q_j . They are called the **complementary** users.
- U_{q_j} : Set of users who have submitted queries that are similar to q_j . They are called the **bridging** users.

The *precise users* have identical interests to user u_i , because they are similar not only in their general interests (as indicated by their membership in the same community) but also in their specific interests (as indicated by their issuing of the same query q_j). Their preferences can be used to promote the accuracy of the results. The *suggestive users* generally have similar interests as u_i but may or may not be specifically in q_j . Their preferences reflect the general interests of the community that serve to broaden the scope of q_j . The *complementary users* indicate *what else* users in the same community are interested in. Finally, the *bridging users* which largely consist of users outside the

user's community reveal different semantical aspects of the query. They lead the user from one semantic aspect (e.g., apple/farming) to another (e.g., apple/computer).

The query-based refined user communities for a particular active user can be determined easily by scanning through U_{u_i} and Q_{q_j} with a linear time complexity. Assume that the number of users in U_{u_i} is $|U_{u_i}|$, and the number of queries in Q_{q_j} is $|Q_{q_j}|$, then the time required to find the refined user communities is $O(|U_{u_i}| + |Q_{q_j}|)$, because we only need to scan through the U_{u_i} and Q_{q_j} once to find the possible refined subclusters from them.

5.2 Click-Based Refined User Communities

A problem with using clickthrough data for personalization is that clickthrough data is unreliable because some users click on many results due to their uncertainty on what they are looking for or complete trust on the ranking of the search engine. Obviously, not all users are helpful to a user even when they belong to the same community. These users are not suitable to be used as a source of high-quality recommendations. Our hypothesis is that users who click on results covering a cohesive set of topics are users who read the result snippets carefully and make their clicks carefully and hence are high-quality users for making recommendations to other users. We use the *concept click entropy* to measure the diversity of a user u 's interest on the result returned from a query q . The concept click entropy $H_{\overline{C}}(q, u)$ [7] of a query q submitted by the user u is defined as follows.

$$H_{\overline{C}}(q, u) = - \sum_{i=1}^t p(\overline{c}_{i_u}) \log p(\overline{c}_{i_u}) \quad (7)$$

where t is the number of concepts clicked by the user u , $\overline{C}_u = \{\overline{c}_{1_u}, \overline{c}_{2_u}, \dots, \overline{c}_{t_u}\}$ is the set of concepts clicked by the user u , $|\overline{c}_{i_u}|$ is the number of times that the concept c_i has been clicked by user u , $|\overline{C}_u| = |\overline{c}_{1_u}| + |\overline{c}_{2_u}| + \dots + |\overline{c}_{t_u}|$, $p(\overline{c}_i, u) = \frac{|\overline{c}_{i_u}|}{|\overline{C}_u|}$.

Since the clicks on the queries are performed by the users after they have read and judged the result snippets with respect to the relevance of the results to their individual needs, the concept click entropy can be used as a mean to identify user behaviors. Thus, we propose the average concept click entropy of a user $H_C(u)$ computed from all their submitted queries $\{q_1, q_2, \dots, q_n\}$.

$$H_C(u) = \frac{1}{n} \sum_{i=1}^n H_{\overline{C}}(q_i, u) \quad (8)$$

K-Means algorithm can then be employed to classify the users into three different classes according to their $H_C(u)$:

- **Focused:** Users with low concept click entropy, i.e., they have very clear topic focuses in the search results and hence only click on a few topics.
- **Average:** Users with higher concept click entropy, and more diversified topical interests than the first user class.
- **Diversified:** Users with high concept click entropy; they click on many results that the search engine returns.

Since the focused users are the best to make high-quality recommendations to other users, we define the *click-based refined user communities* based on the focused users as follows:

- $U^{Focused}$: Set of all *Focused* users.
- $U_{u_i}^{Focused}$: Set of *Focused* users who are similar to u_i .
- $U_{(u_i, q_j)}^{Focused}$: Set of *Focused* users who are similar to u_i and have submitted the query q_j .
- $U_{(u_i, \bar{q}_j)}^{Focused}$: Set of *Focused* users who are similar to u_i and have not submitted the query q_j .

After identifying the refined user communities, clickthroughs of the users from a refined user community are gathered to mine their preferred concepts to provide community-based recommendation. In Section 7.4, we will evaluate and compare the quality of the personalized results with different refined user communities.

6 Community-Based Ranking

To perform community-based personalization for a user community U , we first introduce the preference mining method, SpyNB [5], [7], and then present community-based ranking SVM (*C-RSVM*) to learn a personalized ranking function for U according to the preferences obtained by SpyNB.

6.1 Preference Mining

SpyNB learns user behavior models from preferences extracted from clickthrough data, and assumes that users would only click on documents with titles, abstracts or URLs that are of interest to them. Thus, it is reasonable to treat the clicked documents as positive samples P . On the other hand, the unclicked documents are treated as unlabeled samples U . To predict the reliable negative examples PN from the unlabeled set, each positive sample is used as a “spy” in the Naïve Bayes classifier to get PN from U . Finally, SpyNB assumes that the user would always prefer the positive set rather than the predicted negative set ($d_i < d_j$, $\forall d_i \in P$, $d_j \in PN$).

6.2 Community-Based RSVM Training

We propose community-based ranking SVM (*C-RSVM*) to learn a user community’s concept preferences. A *concept feature vector*, $\phi_C(q, d_k)$, is created for query q and document d_k that q returns. The features and feature weights of $\phi_C(q, d_k)$ are defined below.

Embodied Concepts c_i : If c_i is in the web-snippet of d_k (i.e., c_i is *embodied*, or simply, *contained*, in d_k), the feature weight of c_i in $\phi_C(q, d_k)$, denoted as $\phi_C(q, d_k)[c_i]$, is incremented by one:

$$\forall c_i \in d_k, \phi_C(q, d_k)[c_i] = \phi_C(q, d_k)[c_i] + 1 \quad (9)$$

Related Concepts c_j : In addition to embodied concepts, a concept feature vector $\phi_C(q, d_k)$ also includes features corresponding to concepts that are related to the embodied concepts. This allows documents containing closely related concepts to be matched. Specifically, a concept feature vector includes features derived from the following two types of relations:

- **1. Similarity:** Two concepts which coexist frequently in the search results of query q might represent the same topical interest. If $coexist(c_i, c_j) > \delta_1$ (δ_1 is a threshold), then c_i and c_j are considered as similar.
- **2. Parent-Child Relationship:** More specific concepts often appear with general terms, while the reverse is not true. Thus, if $pr(c_j|c_i) > \delta_2$ (δ_2 is a threshold), we mark c_i as c_j 's child.

From these two relations, four types of related concepts can be obtained: **(1) Similar**, **(2) Ancestor**, **(3) Descendant**, and **(4) Sibling**. Given concept c_i , the feature weights of the related concepts c_j 's are incremented in the concept feature vector $\phi_C(q, d_k)$:

$$\begin{aligned} \forall c_i \in d_k, \phi_C(q, d_k)[c_j] = & \phi_C(q, d_k)[c_j] + sim_R(c_i, c_j) \\ & + ancestor(c_i, c_j) + descendant(c_i, c_j) + sibling(c_i, c_j) \end{aligned} \quad (10)$$

The concept feature vectors, $\phi_C(q, d_k)$'s, and the document preferences of a user community U (see Section 6.1) are used to train a ranking SVM (RSVM) [4] to learn a linear ranking function. In our experiments, an adaptive implementation, *SVMlight* available at [1], is used for the training. It outputs a *community concept weight vector* $\overrightarrow{w_{C,U}}$, which best describes the user community's conceptual interests for rank adaptation of future search results.

7 Experimental Results

In this section, we first evaluate the clustering effectiveness of CADC against four baselines, namely, K-Means, CubeSVD, M-LSA, and BB. We then discover the refined user communities from G_3^C as discussed in Section 5. The refined user communities are then employed in community-based ranking, and the effectiveness of different refined user communities are compared and evaluated.

7.1 Experimental Setup

In the evaluation of traditional information retrieval systems [14], expert judges are employed to judge the relevance of a set of documents (e.g., TREC) based on a description of the information need. The relevance judgment is then considered the standard to judge the quality of the search result. However, the same evaluation method cannot be applied to community-based Web search, because only the user who conducted the search can tell which of the personalized results are relevant to his/her search intent. Thus, the same query issued by two different users may have very different goals. Thus, instead of having expert judges to evaluate the results with optimized information goals, we ask the users to select the specific aspects of the test queries they are interested in. Then, they conduct the search and generate their own clicks. Finally, they make

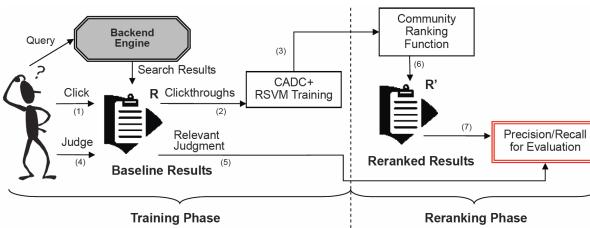


Fig. 3. Flow of the evaluation process

their relevant judgment on the search results with respect to the specific aspects of the queries they had selected. A similar evaluation approach has been used in [13], [8], [9]. To evaluate the performance of CADC and the community-based ranking function, we developed a middleware to serve as Google's frontend for collecting clickthrough data. 100 users are invited to use our middleware to search for the answers of 500 test queries. The 100 users belong to 10 different groups representing 10 different topical interests. The topics are purposely chosen to be very broad (within the top three levels of ODP), providing the users enough room to decide which specific aspects of the query he/she wants to focus on. Each user is assigned with one of the ten different topical interests, and is asked to search 3-8 random test queries within his/her topical interest, and click on the results that he/she considers relevant to the queries within the assigned topic.

Figure 3 shows that flow of the overall evaluation process, which consists of a training phase and a reranking phase (in contrary to the training phase and the testing phase in traditional machine learning evaluation). In the training phase, we provide each user 3-8 test queries together with a topical category to facilitate clickthrough collection. After a query is submitted to the search middleware by a user, a list containing the top 100 search results and the extracted concepts are returned to the user. The user is required to click on the results they find relevant to their queries by looking at the returned web-snippets only. The clickthrough data, which includes the user IDs, the submitted queries and the clicked results, are collected and stored, which is then employed by the five clustering algorithms to produce clusters of similar users, queries, and concepts. 7,392 unique concepts are extracted from the 35,755 unique search results of the 500 test queries for the CCM construction. The concept mining threshold is set to 0.07, which is low enough to include as many concepts as possible in the clustering process. CADC is then employed to cluster CCM to produce sets of similar users as user communities. After discovering the user communities, the clicked results from a user community are treated as positive training samples P in RSVM training. The clickthrough data and the extracted concepts are employed in RSVM training to obtain the community-based ranking function as described in Section 6.

After collecting the clickthrough data, each user was asked to fill in a score (“Good”, “Fair” and “Poor”) for each of the 100 search results returned from each query (by carefully examining the returned 100 web documents one by one) to generate the full *relevance judgment* on the original results R . Documents rated as “Good” are considered relevant documents (i.e. *correct* to the user’s needs), while documents rated as “Poor” are considered irrelevant (i.e. *incorrect* to the user’s needs) and those rated as “Fair” are treated as unlabeled.

Table 2. Precisions of the five clustering methods

	K-Means	CubeSVD	M-LSA	BB	CADC
Precision	0.7275	0.7612	0.8642	0.9141	0.9622
Recall	0.3152	0.2998	0.6064	0.6732	0.7700
F-Measure	0.3733	0.4189	0.6793	0.7531	0.8479

After the training phase, the re-ranking phase begins. The original results R are reranked using the community-based ranking resulted from the training phase to produce the reranked results R' . The ranking of the relevant documents R and R' (before and after reranking) are used to compute *Discounted Cumulative Gain* ($DCG_{100} = \sum_{i=1}^{100} \frac{2^{rel_i}-1}{log_2(1+i)}$, where $rel_i \in \{0, 1\}$ is the relevance value of the position i search result) to measure the effectiveness of our community-based personalization methods among the top 100 search results in Section 7.4. The expectation is that if the ranking function is effective, the unclicked but relevant results are promoted in the reranked list R' , hence increasing the DCG_{100} of R' .

7.2 Optimal Thresholds

To find the optimal terminating/reduction threshold (say t_a^*) for an algorithm a , we repeat the experiment to find the three F-measures $F_{u,t_a,a}$, $F_{q,t_a,a}$, $F_{c,t_a,a}$ for the user, query, and concept clusters of the clustering by setting the terminating threshold $t_a \in [0, 1]$ in 0.05 increments for BB and CADC, and the reduction threshold t_a in 1 increments for KMeans, CubeSVD, and M-LSA. After, the three F-measures are obtained, the average F-measure $AF_{t_a,a}$ of the algorithm a is computed using the following formula:

$$AF_{t_a,a} = \frac{F_{u,t_a,a} + F_{q,t_a,a} + F_{c,t_a,a}}{3} \quad (11)$$

The optimal threshold t_a^* is then obtained when the highest $AF_{t_a,a}$ is achieved ($t_a^* = \text{argmax}_{t_a} AF_{t_a,a}$) for the algorithm a . In Section 7.3, the five algorithms are compared at their own optimal settings (at t_{CADC}^* , t_{BB}^* , $t_{CubeSVD}^*$, t_{M-LSA}^* , and $t_{K-Means}^*$).

7.3 Clustering Effectiveness of CADC

In this Section, we evaluate the effectiveness of CADC in grouping similar objects together. The ground truth clusters are determined according to their predefined topical categories. As shown in [10], clustering algorithms that employ concepts achieve better precisions comparing to content-ignorant methods that consider document only. Thus, all of the five methods compared in this section are based on the *CCM* model.

Table 2 shows the average precisions of K-Means, CubeSVD, M-LSA, BB, and CADC methods obtained at the optimal thresholds (at $t_{K-Means}^*$, $t_{CubeSVD}^*$, t_{M-LSA}^* , t_{BB}^* , and t_{CADC}^* as shown in Section 7.2). K-Means is served as the baseline in the comparison. Hierarchical clustering methods (BB and CADC) are slower, but more accurate comparing to partitional clustering methods, such as CubeSVD and M-LSA. Thus, CADC yields the best average precision, recall, and F-measure values (0.9622, 0.7700, and 0.8479), while BB yields the second best average precision, recall, and F-measure values (0.9141, 0.6732, and 0.7531). The extra divisive phase in CADC helps

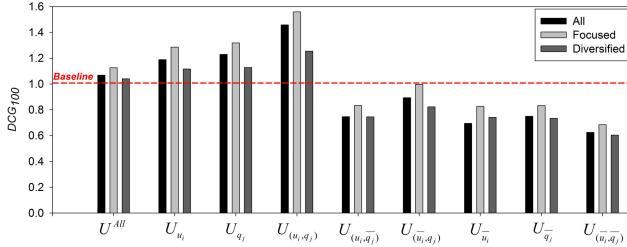


Fig. 4. DCG_{100} for collaborative search with different refined user communities.

to separate dissimilar objects into different clusters, resulting in more accurate clusters comparing to BB.

7.4 Personalized Refinement

In this section, we evaluate the ranking quality of the community-based personalization methods using the clickthroughs from nine different user communities. U^{All} composes of all of the users in the experiment. U_{u_i} , U_{q_j} , $U_{(u_i, q_j)}$, and $U_{(\overline{u_i}, \overline{q_j})}$ are the *Query-based Refined User Communities*. For completeness, we also provide the results for all possible *inverse* user communities (i.e., users whose interests are different from the active user), namely, $U_{(\overline{u_i}, q_j)}$, $U_{(\overline{u_i}, \overline{q_j})}$, $U_{(\overline{q_j}, u_i)}$, and $U_{(\overline{u_i}, \overline{q_j})}$. We further classify the nine user communities into *Focused* or *Diversified* users, which are the *Click-based Refined User Communities* described in Section 5. The community-based rankings are compared against the baseline, which is Google’s ranking. We evaluate the effectiveness of the nine methods by comparing their DCG_{100} achieved with different user communities.

Figure 4 shows the DCG_{100} for the baseline (i.e., the red dotted line) and the nine proposed methods. We observe that only U^{All} , U_{u_i} , U_{q_j} , and $U_{(u_i, q_j)}$ are better than the baseline method, showing that the user communities containing complementing preferences (i.e., $U_{(\overline{u_i}, \overline{q_j})}$, $U_{(\overline{u_i}, q_j)}$, $U_{(\overline{u_i}, \overline{q_j})}$, $U_{(\overline{q_j}, u_i)}$, and $U_{(\overline{u_i}, \overline{q_j})}$) are not good for community-based personalization. We also observe that the personalization methods with *Focused* users always perform better comparing to methods including *All* or *Diversified* users in all of the nine user communities. The $U^{Focused}$ method boosts the DCG_{100} from 1.0683 to 1.1562 (8% in percentage gain) comparing to the U^{All} method. This shows that the focused users can provide useful relevant feedback information for the active user u_i .

We also observe the U_{u_i} method (i.e., the suggestive users) boosts the DCG_{100} to 1.1883 (18% in percentage gain) comparing to the U^{All} method, showing that similar users can always provide useful relevant feedback for the ranking. Moreover, *Focused* suggestive users, $U_{u_i}^{Focused}$, can further boost the DCG_{100} to 1.2861 (27% in percentage gain) comparing to the baseline method. This shows that focused suggestive users provide useful relevant feedback information for the active user u_i . On the other hand, we also observe that U_{q_j} method (i.e., the bridging users) boosts the DCG_{100} to 1.2297 (22% in percentage gain), while the *Focused* bridging users, $U_{q_j}^{Focused}$, can further boost the DCG_{100} to 1.3181 (30% in percentage gain) comparing to the baseline method. This also shows that focused users provide useful relevant feedback information for u_i , and, in particular, focused bridging users also provide useful relevant feedback information for the active query q_j .

The precise users, $U_{(u_i, q_i)}$, are even more closely related with u_i comparing U_{u_i} because they have submitted queries that are similar to q_i as the active user u_i . Thus, we observe that the $U_{(u_i, q_i)}$ method boosts the DCG_{100} to 1.4577 (44% in percentage gain) comparing to the baseline method. If we only consider the focused users in $U_{(u_i, q_i)}$ as $U_{(u_i, q_i)}^{Focused}$, the best $DCG_{100} = 1.5590$ can be obtained (54% in percentage gain comparing to the baseline).

We evaluate also the $U_{(u_i, \overline{q_i})}$ method. The complementary users, $U_{(u_i, \overline{q_i})}$, are similar to u_i because they had issued some queries in common, but not the active query q_i (no prior knowledge about the information retrieved from q_i). We observe that $U_{(u_i, \overline{q_i})}$ is worse comparing to U^{All} , showing that clickthroughs from $U_{(u_i, \overline{q_i})}$ are not quite helpful to the active user u_i . Thus, even if the users U_{u_i} are suggestive, they should have submitted the active query q_j in order to maximize the community-based personalization effect for u_i . Finally, all of the other *complementing* user communities $U_{(\overline{u_i}, q_j)}$, $U_{\overline{u_i}}$, $U_{\overline{q_j}}$, and $U_{(\overline{u_i}, \overline{q_j})}$ are performing badly. Among the four inverse user communities, $U_{(\overline{u_i}, \overline{q_j})}$ performs the worst because it contains nothing related to the active user u_i and the active query q_j . Further, $U_{(\overline{u_i}, \overline{q_j})}^{Diversified}$ performs the worst among all of the community-based personalization methods.

8 Conclusions

A main finding of this paper is that personalization and groupization can be combined to obtain refined user communities to improve a personalized search engine. In this paper, we propose CADC to effectively exploit the relationships among the users, queries, and concepts in clickthrough data. We also investigate various refined user communities to personalize the search results. Experimental results confirm that personalization with refined user communities can provide better search results comparing to the baseline methods. For future work, we plan to investigate the use of more general topics instead of the precise concepts used in CADC to speed up the clustering process.

References

1. svm^{light}, <http://svmlight.joachims.org/>
2. Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. In: Proc. of ACM SIGKDD Conference (2000)
3. Ferragina, P., Gulli, A.: A personalized search engine based on web-snippet hierarchical clustering. In: Proc. of WWW Conference (2005)
4. Joachims, T.: Optimizing search engines using clickthrough data. In: Proc. of ACM SIGKDD Conference (2002)
5. Leung, K.W.T., Lee, D.L.: Deriving concept-based user profiles from search engine logs. IEEE TKDE 22(7) (2010)
6. Leung, K.W.-T., Lee, D.L.: Dynamic agglomerative-divisive clustering of clickthrough data for collaborative web search. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5981, pp. 635–642. Springer, Heidelberg (2010)
7. Leung, K.W.T., Lee, D.L., Lee, W.C.: Personalized web search with location preferences. In: Proc. of ICDE Conference (2010)

8. Leung, K.W.T., Lee, D.L., Lee, W.C.: Pmse: A personalized mobile search engine. IEEE TKDE 99 (2012) (PrePrints)
9. Leung, K.W.T., Lee, D.L., Ng, W., Fung, H.Y.: A framework for personalizing web search with concept-based user profiles. ACM TOIT 11(4) (2012)
10. Leung, K.W.T., Ng, W., Lee, D.L.: Personalized concept-based clustering of search engine queries. IEEE TKDE 20(11) (2008)
11. Pereira Rodrigues, P., Gama, J.: Semi-fuzzy splitting in online divisive-agglomerative clustering. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) EPIA 2007. LNCS (LNNAI), vol. 4874, pp. 133–144. Springer, Heidelberg (2007)
12. Sun, J.T., Zeng, H.J., Liu, H., Lu, Y.: Cubesvd: A novel approach to personalized web search. In: Proc. of WWW Conference (2005)
13. Teevan, J., Morris, M.R., Bush, S.: Discovering and using groups to improve personalized search. In: Proc. of ACM WSDM Conference (2009)
14. Voorhees, E., Harman, D.: TREC Experiment and Evaluation in Information Retrieval. MIT Press, Cambridge (2005)
15. Wang, X., Sun, J.T., Chen, Z., Zhai, C.: Latent semantic analysis for multiple-type interrelated data objects. In: Proc. of ACM SIGIR Conference (2006)

Colored Petri Nets for Integrating the Data Perspective in Process Audits^{*}

Michael Werner

University of Hamburg, Germany

michael.werner@wiso.uni-hamburg.de

Abstract. The complexity of business processes and the data volume of processed transactions increase with the ongoing integration of information systems. Process mining can be used as an innovative approach to derive information about business processes by analyzing recorded data from the source information systems. Although process mining offers novel opportunities to analyze and inspect business processes it is rarely used for audit purposes. The application of process mining has the potential to significantly improve process audits if requirements from the application domain are considered adequately. A common requirement for process audits is the integration of the data perspective. We introduce a specification of Colored Petri Nets that enables the modeling of the data perspective for a specific application domain. Its application demonstrates how information from the application domain can be used to create process models that integrate the data perspective for the purpose of process audits.

Keywords: Business Process Audits, Petri Nets, Business Process Modeling, Process Mining, Business Intelligence, Business Process Intelligence.

1 Introduction

The integration of information systems for supporting and automating the operation of business processes in organizations opens up new ways for data analysis. Business intelligence is an academic field that investigates how data can be used for analysis purposes. It provides a rich set of analysis methods and tools that are well accepted and applied in a variety of application domains, but it is rarely used for auditing purposes.

This article deals with the application of Colored Petri Net models that combine the control flow and data perspective for process mining in the context of process audits. We refer to the example application domain of financial audits for illustration purposes. The benefit of this application domain is the fact that the event data which is necessary for the application of process mining displays structural characteristics that are particularly suitable to be used for the integration of a data perspective. These characteristics relate to the structure of financial accounting and are independent from

* The research results presented in this paper were developed in the research project EMOTEC sponsored by the German Federal Ministry of Education and Research (grant number 01FL10023). The authors are responsible for the content of this publication.

the used source system. The objective of this article is to illustrate an approach for the integration of the data perspective into process models in the context of process mining and process audits. This approach is not restricted to the illustrated example application domain but can be applied in a variety of application scenarios where information about the involved data is available and valuable for process mining purposes.

2 Related Research

Process Mining is a research area that emerged in the late 1990s. Tiwari et al. provide a good overview of the state-of-the-art in process mining until 2008 [1] and Van der Aalst [2] provides a comprehensive summary of basic and advanced mining concepts that have been researched during the last decade. Jans et al. investigate the application of process mining for auditing purposes [3–5]. They provide an interesting case study [6] for compliance checking by using the Fuzzy Miner [7] implemented in the ProM software framework [8]. The study shows how significant information can be derived from using process mining methods for internal audits. The research results are derived from analyzing the control flow of discovered process models and the interaction of users. We are not aware of any implementations or case studies that consider the data perspective in the context of process mining for process audits. One of the reasons may be the fact that the data perspective in process mining has generally not been investigated extensively yet in the academic community [9, 10]. Exceptions are the research results published by Accorsi and Wonnemann [11] and de Leoni and van der Aalst [10]. The research presented by Accorsi and Wonnemann is motivated by finding control mechanisms to identify information leaks in process models. The authors introduce information flow nets (IFnets) as a meta-model based on Colored Petri Nets that are able to model information flows. Instead of using tokens exclusively for the modeling of the control flow colored tokens are used to represent data items that are manipulated during the process execution. De Leoni and van der Aalst use a different approach. Their intention is to incorporate a data perspective for analyzing why a certain path in a process model is taken for an individual case. The modeled data objects influence the course of routing. They introduce Petri Nets with data (DPN-nets) that base on Petri Nets but that are extended by a set of data variables that are modeled as graphical components in the DPN-nets.

3 Application Domain and Requirements

Financial information is published to inform stakeholders about the financial performance of a company. The published information is prepared based on data that is recorded by information systems in the course of transaction processing. Public accountants audit financial statements for ensuring that the financial information is prepared according to relevant rules and regulations. The understanding of business processes plays a significant role in financial audits [12]. The rationale of considering business processes is the assumption that well controlled business processes lead to complete and correct recording of entries to the financial accounts. Auditors

traditionally collect information about business processes manually by performing interviews and inspecting available process documentation. These procedures are extremely time-consuming and error-prone [13]. Process mining allows an effective and efficient reconstruction of reliable process models. Its application would significantly improve the efficiency and effectiveness of financial process audits [14].

Most mining algorithms focus on the reconstruction of control flows in process models that determine the relation and sequence of process activities [9, 10]. Information about control flows is important for auditors to understand the structure of a business process. But this information alone is not sufficient from an audit perspective because an auditor additionally needs to understand how the business processes relate to the entries in the financial accounts [15, 16]. It is therefore necessary to receive information on how the execution of activities in a business process relate to recorded financial entries. This can be achieved by incorporating the data perspective. The relationship between transactions, journal entries and financial accounts is illustrated in Figure 1.

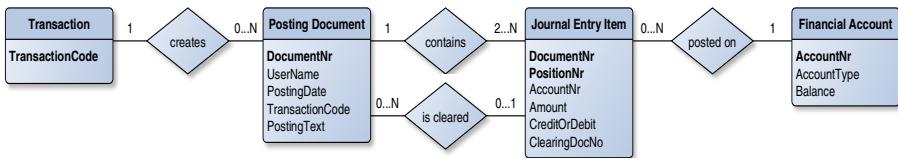


Fig. 1. Accounting Structure Entity-Relationship-Model

A second important concept in financial process audits is materiality [15, 16]. Auditors just inspect those transactions that could have a material effect on the financial statements. To be able to identify which business processes are material information is needed about the amounts that were posted on the different accounts. It is therefore also necessary to model the value of the posted journal entries in the produced process models.

4 Integrating the Data Perspective

The relevant data objects in financial audits are journal entries. They are created during the execution of a business process but their values do not influence the course of routing. They can be interpreted as passive information objects and we therefore refer to the approach used by Accorsi and Wonnemann [11] for integrating them into the process models. Petri Net places and tokens are normally used in process mining to model the control flow. The general approach for integrating the data perspective is to model data objects as colored tokens that are stored in specific places.

The integration is illustrated in the following specification. A Colored Petri Net can formally be expressed as a tuple $CPN = (T, P, A, \Sigma, V, C, G, E, I)$ [17]. Table 1 presents the formal definition of each tuple element, the used net components, and their meaning when applied in the context of this paper. For ease of reference we refer to this type of nets as Financial Petri Nets (FPN) for the remainder of this paper.

Table 1. Specification of Colored Petri Nets for Process Mining in Financial Audits

T is a finite set of transitions				
	The transitions represent the activities that were executed in the process. They display the name of the activity. Further information, for example the transaction code name, can be added.			
P is a finite set of places				
Places in the FPN represent financial accounts and control places.				
Source Place	Sequence Place	Sink Place	Control Places	Control places determine the control flow in a process model. For every process model one source place is modeled that connects to the start transactions. The sink place marks the termination of the process. The control places between the start and end transition determine the execution sequence of the process model. A control place belongs to the set of control places CP.
Account Places				
	credit side of a balance sheet account		debit side of a balance sheet account	The account places represent financial accounts that are affected by the execution of activities in a process. The symbol color indicates the meaning of an account. Account places belong to the set of account places AP.
	credit side of a profit and loss account		debit side of a profit and loss account	
A ∈ P × T ∪ T × P is a set of arcs also called flow relation				
	Control arcs connect control places with transitions. They model the control flow in the model.			
	Posting arcs illustrate the relationship between activities represented as transitions in the model and financial accounts that are modeled as account places.			
	Clearing arcs are used to model that an activity cleared an entry on the corresponding account. Clearing arcs are double-headed arcs and are used as an syntactical abbreviation for two arcs (p,t) and (t,p).			
Σ is a set of non-empty color sets				
colset <i>VALUES</i> = double	The color set contains the possible values that are posted or cleared.			
colset <i>ACCOUNTS</i> = string	The color set contains all account numbers.			
colset <i>ACCOUNTTYPE</i> = boolean	The color set contains {1,0} indicating if the represented account is a balance sheet or a profit and loss account.			
colset <i>CREDorDEB</i> = boolean	The color contains {1,0} indicating if the account place is a representation of the debit or credit side of an account.			
colset <i>EXECUTIONS</i> = int	The color contains {1,...,n} indicating how often a path was chosen in the FPN.			
colset <i>ACCOUNTPLACES</i>	<i>ACCOUNTPLACES</i> is the color set as a product of <i>VALUES</i> * <i>ACCOUNTS</i> * <i>ACCOUNTTYPE</i> * <i>CREDorDEB</i>			

Table 1. (Continued)

V is a finite set of typed variables such that Type[v] ∈ Σ for all variables v ∈ V.
In FPN models arc inscriptions are modeled as constants. Variables are therefore not necessary and $V = \{\}$.
C: P → Σ is a color set function that assigns a color set to each place.
The color set function in FPN assigns different color sets to places depending if they belong to the group of control or account places:
$C(p) \begin{cases} ACCOUNTPLACES & \text{if } p \in AP \\ EXECUTIONS & \text{if } p \in CP \end{cases}$
G: T → EXPR_V is a guard function that assigns a guard to each transition t such that Type [G(t)] = boolean.
EXPR is the set of expressions that is provided by the used inscription language. FPN do not explicitly include guards because they do not model dynamic behavior of transitions that depends on specific input but illustrate the processing of already executed processes. The guard function for FPN is therefore defined as $G(t) = \text{true for all } t \in T$.
E: A → EXPR_V is an arc expression function that assigns an arc expression to each arc a such that Type [E(a)] = C(p)_{MS}, where p is the place connected to the arc a.
The arc expressions in a FPN are constants. The arc expression function assigns to each posting and clearing arc a set of constants that denote the posted or cleared value, the account number, account type and an indicator if it is a credit or debit posting. For each control flow arc the number of execution times is assigned indicating how often this path was chosen in the process model.
$E(a) \begin{cases} \{val \in VALUES, acc \in ACCOUNTS, acctyp \in ACCOUNTTYPE, cord \in CREDorDE \\ \quad \quad \quad \text{with Type } [E(a)] = C(p)_MS = ACCOUNTPLACES \text{ if } p \in AP \\ \{ex \in EXECUTIONS \text{ with Type } [E(a)] = C(p)_MS = EXECUTIONS \text{ if } p \in CP \end{cases}$
I: P → EXPR_Ø is an initialization function that assigns an initialization expression to each place p such that Type [I(p)] = C(p)_{MS}
The initialization function of a FPN assigns initialization expressions to each place as follows:
$I(p) \begin{cases} n' ex \in EXECUTIONS \text{ with Type } [I(p)] = C(p)_MS = EXECUTIONS \text{ if } p = source \\ \emptyset_{MS} \text{ otherwise} \end{cases}$
Only the source place is initialized in a FPN. The initialization expression for $p = source$ generates n tokens in the initial marking $M_0(p)$, one for each connected start transition. The inscription of each token is a member of the set EXECUTIONS.

Figure 2 illustrates a FPN model for a purchasing process. The model includes:

Transitions: $T = \{MB01, MIRO, F110\}$

Places: $P = \{Source, S1, S2, Sink, 100_D, 200_D, 200_C, 300_D, 300_C, 400_D\}$

$CP \subseteq T = \{Source, S1, S2, Sink\}$

$AP \subseteq T = \{100_D, 200_D, 200_C, 300_D, 300_C, 400_D\}$.

Color sets: $VALUES = \{50,000\}, ACCOUNTS = \{100, 200, 300, 400\}, ACCOUNTTYPE = \{0,1\}, CREDorDEB = \{0,1\}; EXECUTIONS = \{1\}; ACCOUNTPLACES = VALUES * ACCOUNTS * ACCOUNTTYPE * CREDorDEB.$

Initialization: $I(p) = \{1'1 \text{ for } p = source \text{ and } \emptyset_{MS} \text{ for } p \neq source\}$

The model shows that the processing of received goods created journal entries with the amount of 50,000 on the raw materials and the goods receipt / invoices receipt (GR/IR) account. The receipt of the corresponding invoice for the purchased goods was processed with activity MIRO which led to journal entries on the GR/IR account and a creditor account. It also cleared the open debit item on the GR/IR account that was posted by MB01. The received invoice was finally paid by executing the activity F110 which posted a clearing item on the creditor account and a debit entry on the bank account.

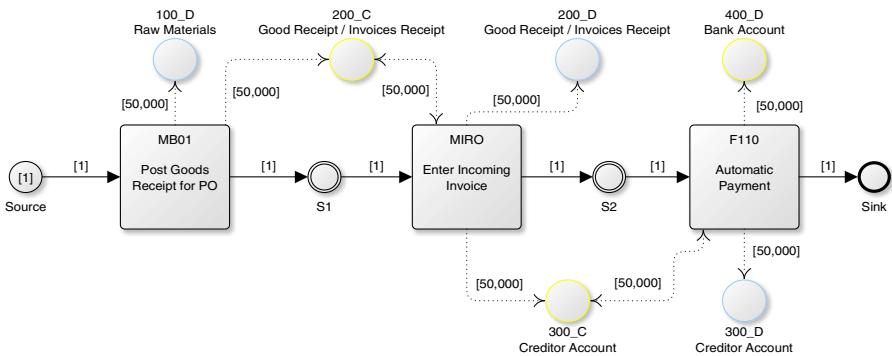


Fig. 2. Simple FPN Example of a Purchase Process

The example in Figure 2 demonstrates how the used Petri Net specification can be used to model the control flow and the data perspective simultaneously in a single model. The transitions in the model create colored tokens when they fire. They store information on the values that are posted on the connected account places. The illustrated model does not only mimic the execution behavior of the involved activities but also the creation of journal entries on the financial accounts. The model shows the execution sequence and the value flows that are produced.

5 Implementation and Experimental Evaluation

We applied the described FPN specification on real world data to evaluate if the theoretical constructs can actually be used in real world settings. The data base for the evaluation included about one million cases of process executions from a company operating in the manufacturing industry. The raw data was extracted from a SAP ERP system and checked for first and second order data defects [18]. We used an adjusted implementation of the Financial Process Mining (FPM) algorithm [19] that is able to produce FPN models. The mining was limited to 100,000 process instances that affected a specific raw materials account. The mining resulted in 113 process variants. They were analyzed in the evaluation phase by observation using the yEd Graph Editor [20] for verifying if the process models presented the desired information adequately. Selected models were further tested using the Renew software [21] for

evaluating if correct FPN were created by simulating the execution of the models. The evaluation demonstrated that FPN can be created correctly by using an adapted FPM algorithm. The produced FPN are able to adequately model the control and data flow based on the used real world data.

The same modeling procedure was used in a different scenario to analyze technical customer service processes which is not described in this paper due to place restrictions.

6 Summary and Conclusion

Process Mining is an innovative approach for analyzing business processes but it is rarely used in the context of process audits. The successful application of process mining requires the consideration of domain specific requirements. A common requirement is the incorporation of a data perspective which has not been addressed extensively yet in the academic community. We have introduced a specification of Colored Petri Nets that allows the modeling of the control flow and data perspective simultaneously. We referred to the application domain of financial audits as a representative example to demonstrate how the data perspective can be included by referring to relevant application domain requirements.

The evaluation of mined process models shows the suitability of the presented specification in real world settings. The evaluation included the data from a SAP system of a single company. It can therefore not be concluded that the results also hold true for other companies or ERP systems. But the presented specification bases on the structure of financial accounting and is therefore independent from any proprietary ERP software implementation or industry. Evaluation results from further current research indicate that the presented results are also applicable in other settings.

References

1. Tiwari, A., Turner, C.J., Majeed, B.: A review of business process mining: state-of-the-art and future trends. *Business Process Management Journal* 14, 5–22 (2008)
2. Van der Aalst, W.M.P.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, Heidelberg (2011)
3. Jans, M., Alles, M., Vasarhelyi, M.: Process mining of event logs in auditing: opportunities and challenges. *Working Paper*. Hasselt University, Belgium (2010)
4. Jans, M.J.: Process Mining in Auditing: From Current Limitations to Future Challenges. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM Workshops 2011, Part II*. LNBP, vol. 100, pp. 394–397. Springer, Heidelberg (2012)
5. Jans, M., van der Werf, J.M., Lybaert, N., Vanhoof, K.: A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications* 38, 13351–13359 (2011)
6. Jans, M., Alles, M., Vasarhelyi, M.: Process Mining of Event Logs in Internal Auditing: A Case Study. In: 2nd International Symposium on Accounting Information Systems (2011)
7. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007. LNCS*, vol. 4714, pp. 328–343. Springer, Heidelberg (2007)

8. Process Mining Group: ProM, <http://www.processmining.org/prom/start>
9. Stocker, T.: Data flow-oriented process mining to support security audits. In: Pallis, G., et al. (eds.) ICSOC 2011 Workshops. LNCS, vol. 7221, pp. 171–176. Springer, Heidelberg (2012)
10. De Leoni, M., van der Aalst, W.M.: Data-Aware Process Mining: Discovering Decisions in Processes Using Alignments (2013)
11. Accorsi, R., Wonnemann, C.: InDico: Information Flow Analysis of Business Processes for Confidentiality Requirements. In: Cuellar, J., Lopez, J., Barthe, G., Pretschner, A. (eds.) STM 2010. LNCS, vol. 6710, pp. 194–209. Springer, Heidelberg (2011)
12. International Federation of Accountants: ISA 315 (Revised), Identifying and Assessing the Risks of Material Misstatement through Understanding the Entity and Its Environment (2012)
13. Werner, M.: Einsatzmöglichkeiten von Process Mining für die Analyse von Geschäftsprozessen im Rahmen der Jahresabschlussprüfung. In: Plate, G. (ed.) Forschung für die Wirtschaft, pp. 199–214. Cuvillier Verlag, Göttingen (2012)
14. Werner, M., Gehrke, N., Nüttgens, M.: Business Process Mining and Reconstruction for Financial Audits. In: Hawaii International Conference on System Sciences, Maui, pp. 5350–5359 (2012)
15. Schultz, M., Müller-Wickop, N., Nüttgens, M.: Key Information Requirements for Process Audits – an Expert Perspective.pdf. In: Proceedings of the 5th International Workshop on Enterprise Modelling and Information Systems Architectures, Vienna (2012)
16. Müller-Wickop, N., Schultz, M., Peris, M.: Towards Key Concepts for Process Audits – A Multi-Method Research Approach. In: Proceedings of the 10th International Conference on Enterprise Systems, Accounting and Logistics, Utrecht (2013)
17. Jensen, K., Kristensen, L.M.: Coloured petri nets. Springer (2009)
18. Kemper, H.-G., Mehanna, W., Baars, H.: Business intelligence - Grundlagen und praktische Anwendungen: eine Einführung in die IT-basierte Managementunterstützung. Vie-weg + Teubner, Wiesbaden (2010)
19. Gehrke, N., Müller-Wickop, N.: Basic Principles of Financial Process Mining A Journey through Financial Data in Accounting Information Systems. In: Proceedings of the 16th Americas Conference on Information Systems, Lima, Peru (2010)
20. yWorks GmbH: yEd - Graph Editor,
http://www.yworks.com/de/products_yed_about.html
21. University of Hamburg: Renew - The Reference Net Workshop,
<http://www.renew.de/>

Former Students' Perception of Improvement Potential of Conceptual Modeling in Practice

Albert Tort, Antoni Olivé, and Joan Antoni Pastor

Department of Service and Information System Engineering
Universitat Politècnica de Catalunya – BarcelonaTech
`{atort,olive,pastor}@essi.upc.edu`

Abstract. Several authors have pointed out a significant gap between Conceptual Modeling (CM) theory and practice. It is then natural that we try to find answers to questions such as: What is the nature of the gap? Which is the magnitude of the gap? Why does the gap exist? and What could be done to narrow the gap? In this paper, we try to answer those questions from the point of view of the former students of a Requirements Engineering and Conceptual Modeling course that have been involved in professional projects. We have surveyed over 70 former students to know how they perceive the degree to which a set of four conceptual modeling artifacts are created in practice, and how they perceive the improvement potential of the creation of those artifacts in practice. For each artifact, we asked a question on the use of the artifact, and one on the recommendation of use of the artifact. We try to identify the reasons why the artifacts were not created, and what would be needed to convince stakeholders and developers to create the artifact, when it is recommended to do it.

Keywords: Conceptual modeling practice, Survey.

1 Introduction

Several authors have pointed out a concern among researchers about a significant gap between Requirements Engineering and Conceptual Modeling (RE/CM) theory and practice [1,2,3]. By theory, we mean here the mainstream methodologies and recommended best practices. If we accept that the gap exists, it is then natural that we try to find answers to questions such as: *What is the nature of the gap?* *Which is the magnitude of the gap?* *Why does the gap exist?* and *What could be done to narrow the gap?*

There have been a few attempts to address those questions both in the CM field and in the broader RE field, but more research is needed to arrive at satisfactory answers [2,3,4,5,6,7].

In this paper, we focus on Conceptual Modeling (CM) practice, and we try to answer those questions from the point of view of the former students of a RE/CM course that are, or have been, involved in projects with a significant CM activity. To this end, we have surveyed over 70 former students to know

how they perceive the degree to which a representative set of CM artifacts are created in practice, and how they perceive the improvement potential of the creation of those artifacts in practice. As far as we know, this is the first time in which former students of an informatics engineering university program are surveyed on the practice of CM.

A comprehensive analysis of CM practice should take into account the types of projects, the activities performed, the methods, techniques and tools used, and the artifacts created. Such an analysis is beyond the scope of this paper. Instead, here we focus on the creation in practice of the set of (closely-related) conceptual modeling artifacts consisting of the use cases, glossaries, structural schemas (or class diagrams) and integrity constraints. The reasons why we chose this focus were that the mentioned artifacts are: (1) widely-recognized as necessary artifacts in one form or another in most conceptual modeling projects; (2) easily identifiable in practice; and (3) well-known by the students.

For each artifact, we asked two main questions: one on the use of the artifact, and one on the recommendation of use of the artifact. Questions on usage are typical in most surveys [8], and in our case the question aimed at knowing the degree to which the artifact is explicitly created in practice. As far as we know, questions on recommendation have not been asked in similar surveys, and in our case the question aimed at knowing whether or not the respondent would have recommended the creation of the artifact when it was not created. We believe that there may be an improvement opportunity of the CM practice when a significant number of respondents would have recommended the creation of an artifact which is well-known by them in the cases in which it was not created.

The structure of the paper is as follows¹. Section 2 briefly describes the RE/CM course taken by the students that later participated in the survey. Section 3 describes how we designed and conducted the survey. Section 4 presents the general results of the survey. Section 5 summarizes the conclusions and points out future work.

2 The RE/CM Course

In order to appreciate the results of the study reported in this paper, in this section we briefly describe the RE/CM course taken by the former students that participated in the survey.

The course started in 2005 as an elective course of the speciality in Software and Information Systems of the five-year program of *Informatics Engineering* taught at the *Barcelona School of Informatics* of the *Universitat Politècnica de Catalunya (UPC) – BarcelonaTech*. Typically, students take the course during their fourth year in the program, after (among others) an introductory course to software engineering.

The main activity of the course is the requirements specification of a software system, including its complete conceptual schema. At the beginning of the course,

¹ See [9] for an extended version of the paper.

the teachers establish a vision within an existing context, which varies each course. The students -working in groups of 5-7 people- have to study the relevant methods, languages and techniques and apply them to the determination and specification of the requirements of a system that realizes the vision.

The groups submit their work in two main deliverables: (1) Requirements Specification and (2) Conceptual Schema. The conceptual schema (written in UML/OCL) must be formally defined using the USE tool [10], and be validated by means of example instantiations. The course emphasizes the artifacts of RE/CM, rather than the process used to develop them.

3 Survey Design and Conduct

We created a web-based survey [11] consisting of seven parts. The first part included two questions aiming at characterizing the number of years of professional experience, and the number of projects with a significant RE/CM activity in which the participant has been involved. Each of the other six parts focused on a specific RE/CM artifact. In this paper we focus only on the four artifacts more closely related to conceptual modeling, which are (alternative names within parentheses): Use cases (scenarios), Glossary, Structural Schema (UML class diagram, ER schema) and Integrity Constraints (UML invariants).

The respondents were asked to answer the questions using a five-point Likert scale, with the values: 1 (*never*), 2 (*rarely*), 3 (*sometimes*), 4 (*often*) and 5 (*always*). The structure of each of the four artifact parts was essentially the same, and consisted of four subparts. The first subpart consisted of only one question \mathcal{U} on the frequency of use of the artifact:

\mathcal{U} : "In general, in the projects in which you have participated, the artifact was created ... ?"

If the answer of the participant to \mathcal{U} was less than 4, then s/he was asked to answer the set of questions of the other three subparts described below. The first was the influence of five causes on the absence of the artifact in the projects in which he participated. In general, the causes suggested were: The methodology used did not require the artifact; it was considered too difficult to create the artifact; stakeholders considered the artifact unnecessary or its cost not justified; there was an implicit definition of the artifact; lack of tools for creating the artifact. There was also an open-ended question for collecting other causes.

The next subpart was a single question \mathcal{R} on the recommendation of use:

\mathcal{R} : "In the projects in which the artifact was not created, would you have recommended its creation, taking into account the situation and the resources available at that time?"

This was a crucial question of the survey, because its answer gives a clear indication about the potential increase of use of the artifact in practice.

The last subpart asked about what would be needed in order to effectively create the artifact in practice. The suggested means were: To know what the artifact is and how to define it; to be convinced that the artifact is needed for system development; better tools for creating the artifact; to be convinced that

Table 1. Participants by number of years and projects (%)

Years	Projects				
	0	1	2	3	>3
≤ 2	1.39	2.78	5.56	0.00	1.39
3	0.00	4.17	6.94	5.56	2.78
4	1.39	0.00	0.00	4.17	8.33
5	2.78	4.17	2.78	1.39	16.67
≥ 6	1.39	2.78	2.78	2.78	18.06
	6.94	13.89	18.06	13.89	47.22

the cost of creating the artifact is worthwhile. There was also an open-ended question for collecting other responses.

We targeted the survey to past students of the indicated RE/CM course. The potential number of survey participants was 369, but we were able to know the current email address of 182 people (49.3%). We sent them an email invitation (and reminders) to visit the survey website. We collected survey responses during October-December 2012.

4 Survey Results and Discussion

In this Section, we describe the general results of the survey. In subsection 4.1 we summarize the number of participants in the survey. Subsection 4.2 provides an assessment of the use of each artifact in their current practice, and subsection 4.3 provides an assessment of the improvement potential in practice of each artifact.

4.1 Participant Characteristics

We received 72 complete responses to our survey, which represents a response rate of 39.6%. Table 1 shows the percentage of participants by the number of years since the course was taken, and the number of projects with a significant RE/CM activity in which the participant has been involved. It can be seen that the 55% of the participants took the course five or more years ago, and that the 61% have participated in three or more relevant projects. We call *most-experienced* respondents to those that have participated in more than three projects.

The table also shows that 6,94% of the respondents have not participated in any project with a significant RE/CM activity. These responses have been ignored in the results reported in this paper.

4.2 Current Practice

The first objective of our work was to obtain an assessment of the use of each CM artifact in practice, as perceived by former students. The assessment can be obtained from the answers to the \mathcal{U} question. We computed the answer average in the Likert scale for each artifact.

Table 2. Current practice by artifact

	All			Most-experienced		
	M	SD	Mdn	M	SD	Mdn
Use Cases	3.10	1.13	3	3.09	1.04	3
Glossary	2.62	1.24	2	2.29	0.99	2
Structural Schema	3.56	1.20	4	3.48	1.18	4
Integrity Constraints	2.64	1.30	2	2.64	1.30	3

Table 2 gives the mean (M), standard deviation (SD) and median (Mdn) for each artifact, for all respondents, and for the most-experienced. It can be observed that there is very little disagreement between the perceptions of all respondents and that of the most-experienced. The largest disagreement is in relation to the glossary.

In order to obtain an assessment of the improvement potential in practice of each artifact, we classified the situations in the current practice into two groups:

- Current Low Practice (\mathcal{CLP}). These are the situations in which the artifact is *never* or *rarely* or *sometimes* used (Likert scale 1, 2 or 3).
- Current High Practice (\mathcal{CHP}). These are the situations in which the artifact is *often* or *always* used (Likert scale 4 or 5).

Formally, the \mathcal{CLP} and \mathcal{CHP} of artifact \mathcal{A} are defined as follows:

$$\begin{aligned}\mathcal{CLP}(\mathcal{A}) &= \frac{\mathcal{U}_1(\mathcal{A}) + \mathcal{U}_2(\mathcal{A}) + \mathcal{U}_3(\mathcal{A})}{\mathcal{U}(\mathcal{A})} * 100 \\ \mathcal{CHP}(\mathcal{A}) &= \frac{\mathcal{U}_4(\mathcal{A}) + \mathcal{U}_5(\mathcal{A})}{\mathcal{U}(\mathcal{A})} * 100\end{aligned}$$

where $\mathcal{U}_i(\mathcal{A})$, $i = 1..5$, is the number of respondents that answered i in the Likert-scale of the \mathcal{U} question of artifact \mathcal{A} , and $\mathcal{U}(\mathcal{A})$ is the total number of respondents to that question.

Our rationale for the classification is that we consider unsatisfactory the \mathcal{CLP} situations because the artifacts are created less than is expected by the theory, while the \mathcal{CHP} situations can be considered satisfactory because the artifacts are created at least *often*.

Figure 1 shows two bars per artifact. The top bar corresponds to all respondents, while the bottom bar corresponds to the most-experienced respondents. Each bar has three segments. The left segment represents the value of \mathcal{CHP} , and the rest of the bar (shown by two segments as will be explained later) represents the value of \mathcal{CLP} .

The artifact with the greatest value of \mathcal{CHP} is the structural schema. The value is similar for both groups of respondents (close to 60%). The artifacts with the least values of \mathcal{CHP} are the glossary and the integrity constraints definition (25%). However, for the most-experienced respondents, the \mathcal{CHP} of glossaries is only the 17%, which is very low.

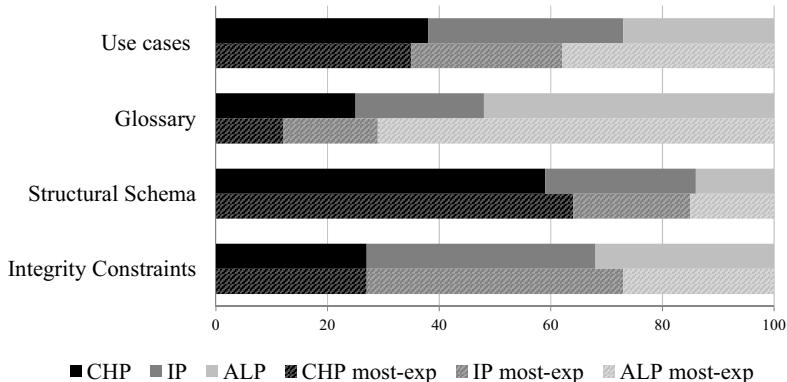


Fig. 1. Current practice and improvement potential of each artifact

The results shown in Fig. 1 provide a partial answer to the questions of *What is the nature of the gap between CM theory and practice?* and *Which is the magnitude of that gap?*:

- An aspect of the nature of the gap is that important CM artifacts are not created in practice as specified by the theory. According to that theory, the artifacts defining the use cases, glossary, structural schema and integrity constraints should be mandatorily created in most, if not all, CM projects, but they are not created in a significant number of them.
- The magnitude of the gap depends on the artifact. The smallest gap is in the structural schema (about 40%). The largest is in the glossary (about 80%). For the other artifacts, the gap lies between these two extremes.

4.3 Improvement Potential

The second objective of our work was to obtain an assessment of the improvement potential in practice of each artifact, as perceived by former students. To this end, we asked to the former students in the \mathcal{CLP} situation the \mathcal{R} question.

Based on the answer to this question, we say that there is:

- A situation with an *Improvement Potential* (\mathcal{IP}) if the answer was *often* (4) or *always* (5), and
- A situation of *Accepted Low Practice* (\mathcal{ALP}) if the answer was *never* (1), *rarely* (2) or *sometimes* (3).

Formally:

$$\mathcal{IP}(\mathcal{A}) = \frac{\mathcal{R}_4(\mathcal{A}) + \mathcal{R}_5(\mathcal{A})}{\mathcal{U}(\mathcal{A})} * 100$$

$$\mathcal{ALP}(\mathcal{A}) = \frac{\mathcal{R}_1(\mathcal{A}) + \mathcal{R}_2(\mathcal{A}) + \mathcal{R}_3(\mathcal{A})}{\mathcal{U}(\mathcal{A})} * 100$$

where $\mathcal{R}_i(\mathcal{A})$, $i = 1..5$, is the number of respondents that answered i in the Likert-scale of the \mathcal{R} question of artifact \mathcal{A} . Note that $\mathcal{CLP}(\mathcal{A}) = \mathcal{IP}(\mathcal{A}) + \mathcal{ALP}(\mathcal{A})$.

Our rationale for the definition of $\mathcal{IP}(\mathcal{A})$ is that we consider that situations have potential for improvement if they are in \mathcal{CLP} but the respondents would have recommended *often* or *always* the creation of the corresponding artifact. That is, if the situation had followed the recommendation, then it would have been in the \mathcal{CHP} situation.

Similarly, our rationale for the definition of \mathcal{ALP} is that we consider that situations remain in an unsatisfactory state if they are in \mathcal{CLP} and the respondents would have not recommended *often* or *always* the creation of the corresponding artifact. That is, if the situation had followed the recommendation, then it would have remained in the \mathcal{CLP} situation.

In Figure 1 the middle segment of each bar shows the value of $\mathcal{IP}(\mathcal{A})$, and the right segment shows the value of $\mathcal{ALP}(\mathcal{A})$.

The improvement potential of the four artifacts (in descending order) is: integrity constraints (41%), use cases (35%), structural schema (27%), and glossary (23%). The results can be considered similar for both groups of respondents.

These results indicate that our former students perceive a large room for improvement of the current practice in each artifact, specially in integrity constraints and use cases. The improvement potential of structural schemas and glossaries is similar, and lower, but their \mathcal{CHP} is quite different.

5 Conclusions

In this paper, we have focused on the recognized gap between CM theory and practice, and we have addressed the questions of: *What is the nature of the gap?* *Which is the magnitude of the gap?* *Why does the gap exist?* and *What could be done to narrow the gap?* To find (at least partial) answers to those questions in our local context, we have surveyed over 70 former university students to know how they perceive the degree to which a set of four CM artifacts are created in practice, and how they perceive the improvement potential of the creation of those artifacts in practice. The artifacts were the use cases, glossary, structural schema and integrity constraints.

We have shown that (one aspect of) the nature of the gap is that important CM artifacts are not created in practice as specified by the theory.

We have shown that the magnitude of the gap depends on the artifact. The smallest gap is in the structural schema, and it is about 40%. The largest is in the glossary, and it is about 80%. For the other artifacts, the gap lies between these two extremes.

We have shown that the improvement potential of the four artifacts (in descending order) is: integrity constraints (41%), use cases (35%), structural schema (27%), and glossary (23%). These results indicate that the former students perceive a large room for improvement of the current situation.

As is usual in similar research works, the results reported in this paper are subject to some threats to their validity beyond our local context, which we

can only summarize here. One is the geographic and domain focus created by drawing the respondents from the former students of an RE/CM course offered by a particular university. Another possible threat is the bias introduced by the form of the questions asked in the questionnaire.

The work reported here can be extended in several directions. Here, we just suggest: (1) taking into account the type and size of the projects and of the companies in which the participants have worked, and (2) analyzing the execution in practice of critical CM activities.

Acknowledgements. This work has been partly supported by the Ministerio de Ciencia y Tecnología and FEDER under project TIN2008-00444/TIN, Grupo Consolidado.

References

1. Neill, C.J., Laplante, P.A.: Requirements engineering: The state of the practice. *IEEE Software* 20(6), 40–45 (2003)
2. Tahir, A., Ahmad, R.: Requirement engineering practices—an empirical study. In: *CiSE* 2010, pp. 1–5. IEEE (2010)
3. Quispe, A., Marques, M., Silvestre, L., Ochoa, S.F., Robbes, R.: Requirements engineering practices in very small software enterprises: A diagnostic study. In: *SCCC* 2010, pp. 81–87. IEEE (2010)
4. Anaby-Tavor, A., et al.: An empirical study of enterprise conceptual modeling. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) *ER* 2009. LNCS, vol. 5829, pp. 55–69. Springer, Heidelberg (2009)
5. Davies, I., Green, P., Rosemann, M., Gallo, S.: Conceptual modelling what and why in current practice. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) *ER* 2004. LNCS, vol. 3288, pp. 30–42. Springer, Heidelberg (2004)
6. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? *Data and Knowledge Engineering* 58(3), 358–380 (2006)
7. Milton, S.K., Rajapakse, J., Weber, R.: Conceptual modeling in practice: An evidence-based process-oriented theory. In: *ICIAFs* 2010, pp. 533–536. IEEE (2010)
8. Hitchman, S.: Object-oriented modelling in practice: class model perceptions in the ERM context. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) *ER* 2000. LNCS, vol. 1920, pp. 397–408. Springer, Heidelberg (2000)
9. Tort, A., Olivé, A., Pastor, J.A.: Former students' perception of improvement potential of conceptual modeling in practice (extended paper). Technical report, Universitat Politècnica de Catalunya (2013), <http://hdl.handle.net/2117/19766>
10. Gogolla, M., Büttner, F., Richters, M.: Use: A UML-based specification environment for validating UML and OCL. *Science of Computer Programming* 69(1-3), 27–34 (2007)
11. Tort, A., Olivé, A., Pastor, J.A.: Survey on requirements engineering and conceptual modeling in practice. Technical report, Universitat Politècnica de Catalunya (2013), <http://hdl.handle.net/2117/19768>

Conceptual Modeling for Ambient Assistance

Judith Michael and Heinrich C. Mayr

Application Engineering Research Group, Alpen-Adria-Universität Klagenfurt, Austria
{judith.michael,heinrich.mayr}@aau.at

Abstract. This paper addresses the conceptual modeling of a person's daily activities, i.e. units of purposeful individual behavior. An integrated set of such models is intended to be used as a knowledge base for supporting that person by an intelligent system when he/she requires so. The work is part of the HBMS¹ project, a research project in the field of Ambient Assisted Living: HBMS aims at supporting people with declining memory by action know-how they previously had in order to prolong their ability to live autonomously at home.

Keywords: Conceptual Behavior Modeling, Ambient Assistance, Behavioral Support, Cognitive Impairments, Activity Theory, User Context.

1 Motivation

With the ongoing acceleration of professional and private life, memories tend to become more transient. Whoever has not already experienced it to overlook a detail or to temporarily forget how to do something: “*What was the sequence of buttons to be pressed for starting a DVD film*”, “*What was I supposed to keep in mind when completing my tax return electronically*”. Immediate assistance is rarely available in situations such as these, or it might be impracticable, too expensive or too imprecise (for example manuals, FAQ lists). This affects particularly elder persons, who experience a growing forgetfulness, and thus increasingly need assistance. Coupled with demographic change, the demand for support services grows exponentially.

An obvious solution lies in support by pervasive computing in as far as this is justifiable and feasible from a psychological, ethical, legal, and technological perspective.

As humans are mobile, support services must be mobile too, adapted to the respective environment and the particular situation. This leads us to the term *Ambient Assistance* [1], describing unobtrusive and, if desired, ubiquitous support. *Ambient Assisted Living* [2] aims particularly at enabling the elderly or persons with impairment to live independent and autonomous lives. Numerous projects deal with the support of healthcare processes through the use of terminal devices, e.g. MARPLE [3], Care-Mate [4]. Other approaches aim at supporting the cognitive performance of an individual in everyday life situations [5]. As an example, Zhou et al. [6] use Case Based Reasoning methods for that purpose; Giroux et al. [7] employ plan recognition.

¹ Human Behavior Monitoring and Support: funded by Klaus Tschira Stiftung GmbH, Heidelberg.

The HBMS - Human Behavior Monitoring and Support Project [8] relies on conceptual behavior modeling. It aims at deriving support services from integrated models of abilities and episodic knowledge an individual had or has temporarily forgotten.

This paper focusses on the modeling aspect of HBMS, concentrating on the description of units of individual target-oriented behavior and their relevant context. It is organized as follows: Section 2 sketches the HBMS aims and architecture, followed by a discussion of various approaches to human behavior modeling (section 3). In section 4 we address the aspects of user context in accordance with [9]. Section 5 introduces the conceptual modeling language HCM-L. The paper ends with a brief report on the experiences gained with a first HBMS prototype, and with an outlook on the next steps for research (section 6).

It is important to us to point out that the HBMS main goal was not to invent just another modeling language. However, from our studies we had to learn that the existing languages do not exactly fit to the needs of modeling human behavior for later support. There are two main reasons: (1) as natural languages continuously are changing along societal development, also modeling languages should be flexibly adaptable for abstracting particular issues. (2) Standardized all-round languages have their merits but often do not provide means for expressing matters to the point efficiently. For instance, following the evaluation of Wohed et al., there are no sufficient solutions in UML Activity Diagrams [10] and BPMN [11] for modeling synchronizing merge patterns, i.e. situations in which a decision relates to a situation earlier in a process.

As a consequence, we endorsed the view of the Open Modeling Initiative OMI [12]: namely to allow for domain specific languages that are tailored for a given application purpose though based on common fundamentals. Such languages then may be lean (few and powerful concepts) and more intuitively used by users from the respective application domain (in our case: psychologists, care givers etc.).

2 HBMS: An Overview

With HBMS we target individuals at any age, who desire a form of support that builds on their own (earlier) practical knowledge. In several workshops and through a survey designed especially for this project, participants requested support in a variety of areas, such as, e.g., the operation of multifunctional devices, participation in eGovernment processes and the use of reminders to help with appointments, taking medication or shopping lists [13]. Fig. 1 shows the development stages in the HBMS Project:

Stage 1: Everyday activities of an individual are observed in a test environment and are textually recorded. From these descriptions conceptual models are derived and integrated into what we call a *Human Cognitive Model (HCM)*.

Stage 2: Observation of behavior patterns is automated by use of sensors or monitor tracking; the data gathered is semi-automatically transformed into a model, which again is integrated into the HCM. During this stage, assistance still has to be requested by the target person, and should adapt itself to the particular situation and context.

Stage 3: Behavior and context (e.g., state of devices, environmental parameters) is observed automatically if wanted, and support is offered automatically (by comparing currently observed behavior from HCM content), where needed. This final phase, however, does not lie within the current scope of the project.

HBMS is an interdisciplinary project, as it not only looks at the informatics-related issues of elicitation, modeling, management, extraction, and representation, but also aims to explore and clarify psychological, technical, legal, and ethical considerations.

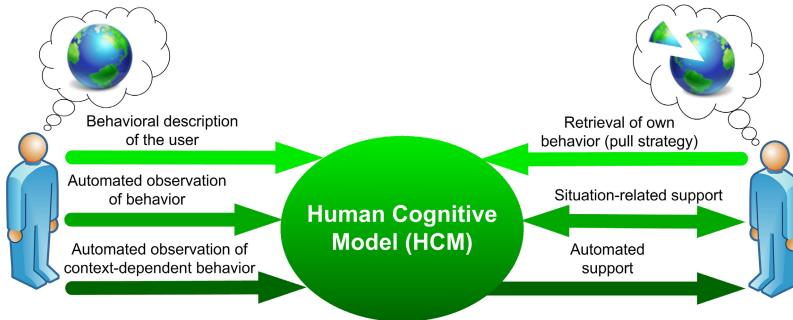


Fig. 1. The development stages in the HBMS Project

Other publications emerging from the HBMS project look closely at the overall project concept and the target groups [8], explore the project architecture and the results of an empirical study carried out with potential users [13], describe the integration of behavioral models [14], discuss quality management aspects [15], and report on a comprehensive control pattern-based analysis of our modeling language [16].

3 Approaches to Modeling Human Behavior

The modeling of human behavior and actions has been studied in several disciplines for understanding and predicting human behavior (including, a military perspective, e.g. [17]). Behavioral modeling has also been addressed in the context of so-called “synthetic agents”, e.g. [18]. Researchers in the field of Psychology use behavioral models in their exploration of processes of judgment and decision-making.

Activity Theory forms an essential foundation [19]. It involves observing the nature of human activities on three levels: The level of the *activity* (the overall process), the level of the *action* (subtasks) and the level of *operations* that realize actions. While activities are informed by *need*, individual actions each pursue a specific *goal*. As the actions meet with success, the need of the overall activity is extinguished. In order to put these actions into effect, individual operations are performed.

Activity Theory rests on five principles: the *hierarchical structure of activities*, *object orientation*, *mediation*, *continuous development*, and *differentiation between internal and external activities*. The suitability of this theory for Human Computer Interaction (HCI) was successfully demonstrated in [20]. In further studies, the theory is also used to model the contexts of specific situations [21].

The modeling concepts applied can vary enormously depending on the respective goal. For example, [7] lists so-called “lattice-based models”, supplemented with probabilities, Bayesian networks, Petri networks, rule-based approaches, and ad-hoc modeling. [22] uses Semi-Markov models to identify Activities of Daily Living (ADLs).

The focus of HBMS is on supporting individuals with their own concrete, episodic knowledge (memory of experiences and specific events). For this purpose, the modeling concepts referred to above are only of limited use, as they can be used to describe actions and their sequences, but do not capture the situational context in detail: for example, the remote control of a DVD recorder, and its precise layout.

Conceptual modeling languages such as the Unified Modeling Language UML provide these possibilities, as long as they cover the modeling dimensions [23]: Structure view (e.g. UML class diagram), functional view (e.g. UML methods), and dynamic view (e.g. UML activity diagrams or state charts). Conceptual business process modeling languages focus on series of actions and generally do not include own concepts for the modeling of structure views; instead they refer to those of the UML. The Business Process Modeling Notation BPMN serves as an example [24]. Despite their expressive power, these languages have weaknesses as has been sketched in section 1.

Our goal is a lean and simple language focused on modeling human behavior that adopts proven concepts from existing languages and waives unneeded ones. The concepts were developed by building upon the experiences gained with KCPM [25], a user-centered language for requirements modeling.

4 User Contexts

Human behavior is determined by more than merely the activities themselves. It is important to consider the context, within which the respective person is moving. In [9] the user context is divided into separate areas: the *task context*, the *personal context*, the *environmental context*, the *social context*, and the *spatio-temporal context*.

The main focus of HBMS is on the *task context*: Everyday activities of a person have to be modeled in detail and in all observed variants, their motives and goals; e.g. to reach the goal of watching a particular DVD film: take the DVD and TV remote control, press the resp. ON buttons, select the desired function buttons etc.

The *personal context* of an individual refers to information about mental and physical parameters including handicaps. This information will be valuable during the ‘productive’ phase, i.e. when providing support to a person with declining cognitive capacity: to choose the best medium and form for help presentation, or to trigger an alert to a relative or medical professional in case of observed degradation. Clearly, this has to comply with all associated ethical and data protection issues.

The *environmental context* refers to the environment of a user, for example: persons and things, with which one communicates or interacts like the remote control.

The *social context* comprises the social environment of the target person: information about friendships, relatives or colleagues. For the purpose of HBMS, such persons will be ‘modeled’ if they are linked to certain activities. Should future users’ needs require broader types of social relations, our approach will be adapted.

The *spatio-temporal context* draws upon information about time, frequency, duration of activities, locations, and movements.

Fig. 2 depicts the role of context in HBMS. A *Behavioral Unit Model (BUM)* represents an integrated view on all *observed action sequences* of a particular activity. BUM's are grouped in topical *clusters*. BUM's, clusters and the sequence models of observed actions together form the task context.

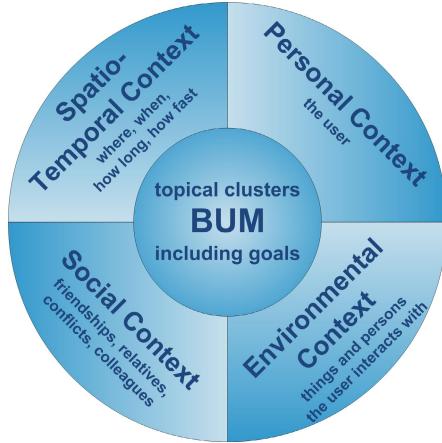


Fig. 2. Context models in HBMS

5 The Modeling Language HCM-L

The aim of HBMS is to support a person on the basis of a model of his/her own behavior. Thus, the HBMS modeling language focuses on concepts for modeling the (sequences of) actions of a person and their contexts in detail. Such sequences are not necessarily identical, even when sharing the same objective: they can vary, e.g., in their order or due to the omission of certain action steps. Consequently, the modeling language must offer concepts for abstracting action variations such that subsequently all possible variations can be derived (as instances). This is analogous to so-called *case prototypes* in case-based reasoning [26] functioning as abstractions of all related cases, and corresponds to “*product lines*” in software engineering [27].

The next sections describe the key concepts of our *Human Cognitive Modeling Language HCM-L* and their most important (meta) attributes; following [28], the HCM-L syntax is described by a meta model (fig. 3), the semantics by explanation and the notation by a set of graphical elements.

5.1 Behavioral Units and Activities

The key concept of HCM-L is that of *behavioral unit*. It is defined as an aggregate of *operations* which together lead to reaching a *goal* in daily life (see *Basic* [5] or *Instrumental* [29] *Activities of Daily Living*). Thus, a behavioral unit corresponds to an

action as defined by Activity Theory, and to a “use-case” in business process modeling, e.g., fulfilling the goal to watch a particular DVD film. Typically the operations of such action form a sequence; this is captured by the concept *flow*: *outgoing* from the predecessor operation and *incoming* for the successor. Since a behavioral unit’s goal may be reached by different sequences, the unit may have one or more *possible beginning* (meta attribute) and, similarly one or more *successful ending*. The *goal distance* indicates the length of the shortest path to a successful ending.

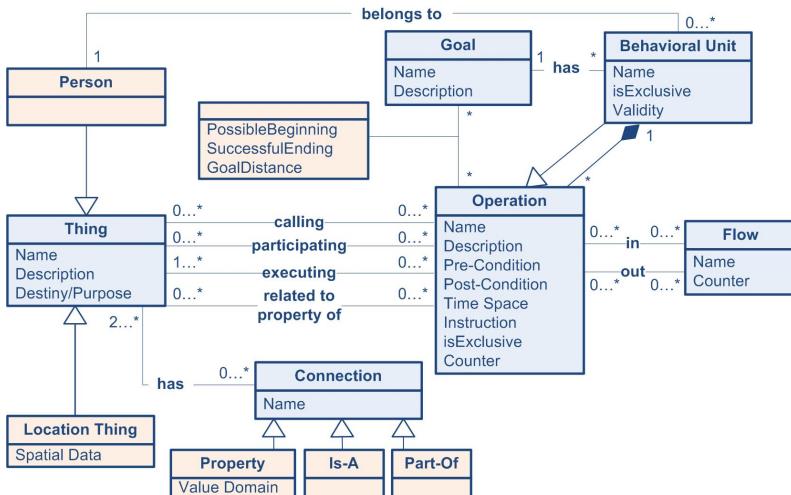


Fig. 3. Meta schema of the HCM-L definition with key attributes

The granularity chosen will depend on the prevailing circumstances. For example, one could perceive the tying of shoe laces as a behavioral unit, but this could also be viewed as an operation of a broader unit of “putting shoes on”, which again could be seen as an operation pertaining to a unit “getting dressed”. In other words, behavioral units themselves can be regarded as operations. This is supported in HCM-L by the generalization relationship between behavioral unit and operation.

To be executable, an operation may depend on a (possibly complex) *pre-condition*. For example, the DVD player must be switched on, before a DVD can be loaded. Similarly, the execution of an operation may set a *post-condition* that, e.g., has consequences on the subsequent flow. These conditions refer to properties, time and space circumstances. In particular they occur with process branching and merges.

It might surprise that we do not model conditions separate from the operations like e.g. in Petri nets or event process chains. This emphasizes our focus on operations, the sequences of which become transparent in lean diagrams (see Fig. 5). Clearly, these could easily be expanded into bipartite graphs separating conditions and operations.

Instruction defines the functional semantics of an operation. For situations where an operation or even a whole behavioral unit should or cannot be interrupted by another one, HCM-L provides the meta attribute *isExclusive*.

Operations can (or must not) occur at specific points in or periods of time, and they may have a duration. This is captured by the attribute *Time Space*. The formal language for specifying conditions, time spaces and value domains is work in progress and reflects the research presented in [30]. It is clear, however, that, regardless of the selected language, the limits of intuitive comprehension are soon reached (see above), as most individuals have only learned to handle simple logical linkages. Realistic modeling and, automated model creation nevertheless require such a language.

5.2 Things and Connections – Elements in the World and Their Relationships

For modeling the context of actions, HCM-L adopts the concepts *thing* and *connection* from KCPM [25]. Things describe arbitrary concrete or abstract objects, also persons, connections model relationships between things.

Every subject and object has a *destiny* (in the sense of the purposes it serves). This meta attribute will be important for support: using a thing against its intension (e.g., a comb to brush the teeth) may induce starting help.

The person to be supported has to be modeled her/himself (concept *person*). Thus, the modeled behavior can be associated to this person (association *belongs-to*).

The concept of *Location Thing* was introduced to allow capturing spatial data as precise as necessary (e.g. coordinates, temperature, humidity, noise etc.); e.g., it is important to know where the remote control was deposited after last used. The meta attributes of the location thing concept are based on [6], where these attributes are compiled from sensor data and recorded as vectors in a case-base.

Operations are related to things: there are *calling* things, which initiate operations, *participating* things, which contribute to or are manipulated by operations, and *executing* things, which perform operations.

The connection concept has specializations to (1) *property* relations such as “the remote control has a display (thing)”, that can show the current channel (*value domain*); (2) Aggregation and decomposition (*part-of*), and (3) specialization and generalization (*is-a*). A thing is called *attribute*, if it is target of a property.

5.3 Graphical Representation

The graphical representations of the few HCM-L concepts are shown in Fig. 4. They follow the principles for designing effective visual notations presented in [31]. In line with these, behavioral units and their context can be modeled and reproduced in a combined form.

Figure 5 shows a simplified version of the operations and flows of the behavioral unit ‘evening activity’. The model has been developed and drawn using the HCM Modeling tool which was developed using the ADOxx® platform [32], and equipped with a simulation interface for visualizing concrete operation flows. Since the picture should be self-explanatory, we only hint at two specifics of HCM-L:

‘watch a DVD’, ‘watch TV’ and ‘read a book on the e-reader’ are aggregations (operations with sub-operations). When clicking on the ‘+’ symbol the resp. sub-model is shown;

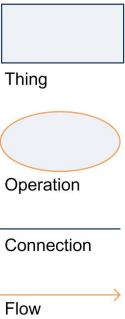
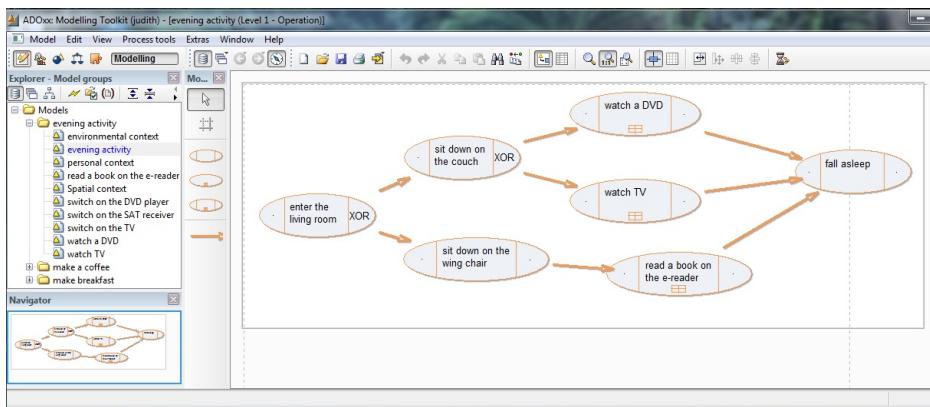
Basic Elements	Thing	Connection	Operation
 Thing Operation Connection Flow	 Person Location Attribute	 Is-A Part-Of Property Calling Participating Executing	 Operation with Pre- and Post-Condition Expression Operation with suboperations

Fig. 4. HCM-L modeling symbols

Post-conditions may be headed by the logical operations AND, OR and XOR leading to flow forks with the resp. semantics. Pre-conditions may be headed by AND (regular merge), XOR (simple merge) and SOR ('synchronized or'). The latter always relates to a Multi-Choice construct occurring earlier in the actual instance of the behavioral unit and denotes a wait until all incoming branches have been performed. For more details see [16].

**Fig. 5.** Operations and flows of a behavioral unit 'evening activity'

Unfortunately, there is not space in this paper to show also a picture of the complete context model or further screen-shots of the model details. The interested reader is referred to the HBMS website <http://hbms-ainf.aau.at/>.

6 First Experiences and Future Perspectives

To check the completeness of HCM-L for the intended use, a pattern-based analysis [16] has been performed on the basis of the Workflow Pattern framework (www.workflowpatterns.com).

Another check was the compliance with Activity Theory [19]. Unfortunately, there is not enough space to report on the (positive) results in detail.

For a proof-of-concept we developed a HCM Modeling and Support Tool. It provides a graphical interface for modeling [33] and the possibility to present derived support information on a handheld device [13]. This prototype was firstly tested on 40 individuals (mostly ages 50+) on the occasion of the “Long Night of Research, Austria” event, using as an example the preparation of a cup of coffee. The respective operations of each individual were modeled, and then rendered as a set of step-by-step instructions on a smartphone or a tablet PC. The visitors regarded this kind of support system, both for themselves and for their environment, as meaningful and useful in everyday life; we received a consistently positive feedback. Some people wanted to take the prototype home immediately, and putting it to use.

Currently, we run an evaluation with 60 people in 3 age groups to assess the support mechanisms concerning the textual, graphical and multimedia representation. The experiences made, and the comments and suggestions gathered on these occasions have resulted in adaptions/extensions of the meta model leading to the version presented within this paper.

The next step will be a real life experiment with a particular test person.

References

1. Arezki, A., Monacelli, E., Alayli, Y.: Ambient Assistance Using Mobile Agents. In: Proc. of the First Int. Conf. on Smart Systems, Devices and Technologies, pp. 89–95 (2012)
2. Steg, H., et al.: Europe Is Facing a Demographic Challenge. In: Ambient Assisted Living Of-fers Solutions. VDI/VDE/IT, Berlin (2006)
3. Pryss, R., Tiedeken, J., Kreher, U., Reichert, M.: Towards Flexible Process Support on Mobile Devices. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBP, vol. 72, pp. 150–165. Springer, Heidelberg (2011)
4. Baumeister, J., Reutelshofer, J., Puppe, F.: KnowWE: A SemanticWiki for Knowledge Engineering. In: Applied Intelligence, vol. 35(3), pp. 323–344 (2011)
5. Katz, S.: Assessing self-maintenance: Activities of daily living, mobility, and instrumental activities of daily living. Journal of the Am. Geriatrics Society (31), 721–727 (1983)
6. Zhou, F., et al.: A Case-Driven Ambient Intelligence System for Elderly in-Home Assistance Applications. Institute of Electrical and Electronics Engineers, New York (2011)
7. Giroux, S., et al.: Pervasive behavior tracking for cognitive assistance. In: Proc. of the Int. Conf. on Pervasive Technologies Related to Assistive Environments. ACM, NY (2008)
8. Griesser, A., Michael, J., Mayr, H.C.: Verhaltensmodellierung und automatisierte Unterstützung im AAL Projekt HBMS. In: Proc. AAL 2012, Berlin (2012)
9. Kofod-Petersen, A., Mikalsen, M.: Context: Representation and Reasoning, Special issue of the Revue d’Intelligence Artificielle on "Applying Context-Management" (2005)

10. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N.: Pattern-Based Analysis of the Control-Flow Perspective of UML Activity Diagrams. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 63–78. Springer, Heidelberg (2005)
11. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N.: On the Suitability of BPMN for Business Process Modelling. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 161–176. Springer, Heidelberg (2006)
12. Karagiannis, D., Grossmann, W., Höfferer, P.: Open Model Initiative: A Feasibility Study. University of Vienna, Dpmpt. of Knowledge Engineering (2002), <http://www.openmodels.at>
13. Michael, J., Grießer, A., Strobl, T., Mayr, H.C.: Cognitive Modeling and Support for Ambient Assistance. In: Kop, C. (ed.) UNISON 2012. LNBP, vol. 137, pp. 96–107. Springer, Heidelberg (2013)
14. Michael, J., Bolshutkin, V., Leitner, S., Mayr, H.C.: Behavior Modeling for Ambient Assistance. In: Proc. Int. Conf. on Management and Service Science (MASS), Shanghai (2012)
15. Shekhovtsov, V., Mayr, H.C.: A Conceptualization of Quality Management Functionality in Cognitive Assistance Systems (submitted for publication)
16. Mayr, H.C., Michael, J.: Control pattern based analysis of HCM-L, a language for cognitive modeling. In: Proc. ICTer 2012, pp. 169–175. IEEE (2012)
17. Silverman, B.G., et al.: Toward A Human Behavior Models Anthology for Synthetic Agent Development. In: Proceedings of the Conference on Computer Generated Forces and Behavioral Representation. SISO (2001)
18. Zacharias, G., MacMillan, J., Van Hemel, S.B. (eds.): Behavioral Modeling and Simulation: From Individuals to Societies. The National Academies Press (2008)
19. Leont'ev, A.N.: Activity, Consciousness, and Personality. Prentice-Hall (1978)
20. Bannon, L., Bødker, S.: Beyond the interface: Encountering artifacts in use. In: Carroll, J. (ed.) Designing Interaction: Psychology at the Human-Computer Interface. Cambridge University Press, Cambridge (1991)
21. Kofod-Petersen, A., Cassens, J.: Using Activity Theory to Model Context Awareness. In: Roth-Berghofer, T.R., Schulz, S., Leake, D.B. (eds.) MRC 2005. LNCS (LNAI), vol. 3946, pp. 1–17. Springer, Heidelberg (2006)
22. Clement, J., Ploennigs, J., Kabitzsch, K.: Smart Meter: Detect and Individualize ADLs. In: Proc. AAL 2012, Berlin (2012)
23. Hesse, W., Mayr, H.C.: Modellierung in der Softwaretechnik: eine Bestandsaufnahme. Informatik-Spektrum 31(5), 377–393 (2008)
24. Allweyer, T.: BPMN 2.0 Introduction to the Standard for Business Process Modeling. BoD – Books on Demand (2009)
25. Kop, C., Mayr, H.C.: Conceptual Predesgin - Bridging the Gap between Requirements and Conceptual Design. In: Proc. ICRES 1998. Colorado Springs (1998)
26. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7, 39–59 (1994)
27. Pohl, K., Böckle, G., van der Linden, F.J.: Software product line engineering: foundations, principles, and techniques. Springer (2005)
28. 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)
29. Lawton, M.P., Brody, E.M.: Assessment of older people: Self-maintaining and instrumental activities of daily living. Gerontologist 9, 179–186 (1969)

30. Olivé, A., Raventós, R.: Modeling events as entities in object-oriented conceptual modeling languages. *Data & Knowledge Engineering* 58, 243–262 (2006)
31. Moody, D.: The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Trans. Software Eng.* 35, 756–779 (2009)
32. Karagiannis, D.: Business Process Management: A Holistic Management Approach. In: Kop, C. (ed.) UNISON 2012. LNBIP, vol. 137, pp. 1–12. Springer, Heidelberg (2013)
33. Bolshutkin, V., Steinberger, C., Tkachuk, M.: Knowledge-Oriented Approach to Requirements Engineering in the Ambient-Assisted Living Domain. In: Kop, C. (ed.) UNISON 2012. LNBIP, vol. 137, pp. 205–207. Springer, Heidelberg (2013)

Empirical Evaluation of the Quality of Conceptual Models Based on User Perceptions: A Case Study in the Transport Domain

Daniela S. Cruzes, Audun Vennesland, and Marit K. Natvig

SINTEF, NO-7465, Trondheim, Norway

{daniela.s.cruzess, audun.vennesland, marit.k.natvig}@sintef.no

Abstract. Today, many companies design and maintain a vast amount of conceptual models. It has been also observed that such large model collections exhibit serious quality issues in industry practice. A number of quality frameworks have been proposed in the literature, but the practice is that practitioners continue to evaluate conceptual models in an ad-hoc and subjective way, based on common sense and experience. Therefore, there is a lack of empirical works in the evaluation of conceptual frameworks. This paper reports an empirical qualitative study on the evaluation of the quality of a conceptual framework in the domain of transport logistics, using existent quality evaluation frameworks. The results show how the users perceive the ease of understanding, the usefulness, the perceived semantic quality and satisfaction with the models included in the conceptual framework. The results also provided their view on advantages, challenges and improvements to be performed in the framework.

1 Introduction

Conceptual modeling is the process of formally documenting a problem domain for the purpose of understanding and communication among stakeholders [5]. Conceptual models are central to IS analysis and design, and are used to define user requirements and as a basis for developing information systems to meet these requirements. More generally, they may be used to support the development, acquisition, adaptation, standardization and integration of information systems [5]. While a good conceptual model can be poorly implemented and a poor conceptual model can be improved in later stages, all things being equal, a higher quality conceptual model will lead to a higher quality information system [5].

Today, many organizations design and maintain a vast amount of conceptual models. It has been observed that such large model collections exhibit serious quality issues in industry practice [6,14], but acceptable evaluation criteria and methods are yet not well established. As a young, multi-disciplinary field, conceptual modeling still lacks empirical studies on the research methodologies and evaluation criteria for the conceptual models (CM) or reference models that have been created for the different domains [25]. The challenge one faces is that, on the one hand, the scientific perspective demands precise, consistent and complete reference models, but on the other hand, these models should be easy to implement and understand [25].

In addition, the conceptual model end-users, and more generally stakeholders, have their perception shaped by their own experience, as well as by socio-cultural factors, and they will be the ones that can evaluate the quality of a model [23]. Even when the modelers use well-established models and techniques, clarity and effectiveness of these models may depend more on things not stated in the spec than on the official notation of the models. Method and style subjectively used by the modeler will consequently influence the final quality of the conceptual model from a user perspective.

A number of quality evaluation frameworks have been proposed in the literature [7][9][10][23][28], but the practice is that practitioners continue to evaluate conceptual models in an ad-hoc and subjective way, based on common sense and experience reports [13][25]. The Lindland et. al [9] framework suggests that a systematic evaluation of quality considers a model's syntax (how well does the model adhere to the rules of the modeling language), semantics (how well the model reflect the reality modeled) and pragmatic (how well is the model understood and used). It is generally agreed that IS stakeholders rely on their perception of reality in order to evaluate semantic quality. Krogstie et al [8] extended the Lindland et al. evaluation framework with a fourth quality type, namely perceived semantic quality, which is described as the correspondence between the information that users think the model contains and the information that users think the model should contain, based upon their knowledge of the problem domain. Wand and Weber based on Bunge's also proposed a ontological theory (the Bunge–Wand–Weber representational model: BWW) [28] focusing on the process of conceptual modeling. Maes and Poels [10][11] further extended these evaluation frameworks by focusing on the different user perceptions of CM quality since it is the user's perception of quality that will determine the effectiveness of the conceptual model. Recently, Nelson et. al. [23] proposed a comprehensive Conceptual Modelling Quality Framework, bringing together Lindland's et al, evaluation model and Wand and Weber based on Bunge's ontology, but do not define the metrics in which a empirical study can be built upon to evaluate a specific conceptual model.

In this paper, we build upon these approaches, especially on Maes and Poels set of questions and metrics, in order to create a qualitative interview guide to collect the perceived quality of a conceptual framework, based on users perceptions of usage of the conceptual framework. We report a study on the evaluation of quality of a conceptual framework in the domain of transport logistics. Our conceptual framework, named here CF¹, has a goal to allow different stakeholder groups in the transport logistics domain at all skill levels to collaborate around a common view of the processes that exist within the domain. Therefore we needed to have multiple levels of models and styles in order to address the separate goals and concerns of business users, analysts and architects, and software developers, at the same time, we have challenges on keeping the understandability of the model in a level that it is feasible for the model sustainability. These and other aspects of the model were discussed in the evaluation of the model and described in this paper.

¹ This framework started out as ARKTRANS. ARKTRANS has since been further developed in Europe to the Common Framework for ICT in Transport and Logistics (Common Framework in short). <http://www.its.sintef9013.com/CF/v01/>

This paper is organized as follows. The next section describes the background of the literature related to evaluation frameworks to conceptual models. Section 3 describes the design of the study and the context of the case study. Section 4 describes the results of the study and finally, in the conclusions we discuss the implications of the study for research and practice.

2 Background

Maes and Poels [10][11] evaluation framework focuses on the different user perceptions of CM quality since it is the user's perception of quality that will determine the effectiveness of the conceptual model. This is the model that we adopted to drive our research and the questions in the questionnaire. The reason is two-fold: 1) it is aligned with the technology acceptance theories which are used to explain how users come to accept a specific technology and IS success models [15][16][17][18], and 2) the authors provide a validated measurement instrument that is solely based on conceptual model end-user evaluations, this is important because as evidence shows, the user perceptions of the CM quality are important determinant of the users' satisfaction with the model [8][10].

Figure 1 shows the main components and theirs relationships on the Maes and Poels evaluation model [10]. The authors stress that a discussion of the quality of conceptual models from the users' point of view is relevant. By adapting measures stemming from popular information system success models to the area of conceptual modeling, they demonstrate that beliefs such as perceived ease of understanding and perceived semantic quality influence various attitudes such as perceived usefulness and eventually user satisfaction.

The perceived semantic quality (PSQ) represents a success measure at the semantic level of information and it concerns how well the intended meaning is being conveyed to and interpreted by the receiver. Users assess how well the model serves its intended purpose. Maes and Poels states that this information should be accurate, complete, up to date and presented in a format that advances users understanding of the underlying reality when performing a certain task [11]. Lindland et al. [9] includes correctness, completeness and consistency as quality properties for the conceptual models, we also added these aspects to our evaluation.

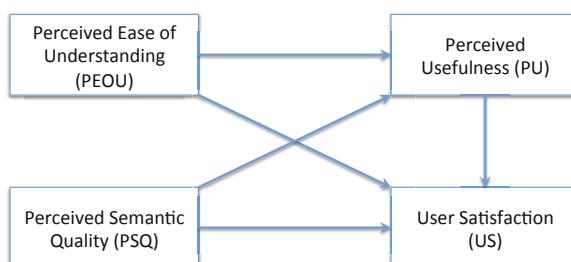


Fig. 1. User evaluations based quality model for Conceptual Modeling Scripts [11]

Table 1. Measure for the Evaluation of Conceptual Models (CM) [10]

PEOU₁	It was easy for me to understand what the CM was trying to model.	PU₁	Overall, I think the CM would be an improvement to a textual description of business process (BP).
PEOU₂	Using the CM was often frustrating.	PU₂	Overall, I found the CM useful for understanding the process modeled.
PEOU₃	Overall, the CM was easy to use.	PU₃	Overall, I think the CM improves my performance when understanding the process modeled.
PEOU₄	Learning how to read the CM was easy.	PSQ₁	The CM represents the business process correctly.
US₁	The CM adequately met the information needs that I was asked to support.	PSQ₂	The CM is a realistic representation of the business process.
US₂	The CM was not efficient in providing the information I needed.	PSQ₃	The CM contains contradicting elements
US₃	The CM was effective in providing the information I needed.	PSQ₄	All the elements in the CM are relevant for the representation of the BP
US₄	Overall, I am satisfied with the CM for providing the information I need-ed.	PSQ₅	The CM gives a complete representa-tion of the business process

The perceived ease of understanding (PEOU), is based on how the users of the model will evaluate how easy it is to read the information in the model and interpret it correctly. Maes and Poels define the ease of using a CM as the degree to which a person believes that using a conceptual modeling script for understanding the problem domain and IS requirements would be free of mental effort.

The perceived usefulness (PU) is an important concept for measuring the users' overall quality evaluation of a model, since the actual objective of using a conceptual model can have a variety of external influence factors. The perceived usefulness is defined as the degree to which a person believes that using a particular model has enhanced his or her job performance.

The user satisfaction (US) is the subjective evaluation of the various consequences evaluated on a pleasant-unpleasant continuum. A general evaluation towards the use of a conceptual model can be measured in terms of how satisfied users are with the model with respect to its purpose. More specifically, with the extent to which users believe the CM meets their informational needs.

In order to evaluate the components of the model, Maes and Poels used some well-validated measures for the PSQ, PEOU, PU and US constructs. The general measurement items for each of the four constructs are shown in Table 1, they were evaluated in a series of experiments [10], we adapted them into a structured interview as shown in Appendix A.

3 Methodology

Addressing the aspects discussed in the previous sections we performed an exploratory empirical case study on quality of a conceptual framework for transport logistics

based on user perceptions. This study adapts the instruments developed by Maes and Poels [10][11] to a semi-structured interview guide, maintaining the same metrics and aspects of evaluation as the original authors (See Appendix A).

The interviews were conducted with users of this conceptual framework, named here CF. The aim is to investigate the user satisfaction with the framework as well as challenges, advantages and improvements that should be addressed in the framework. We also reflect on the lessons learned from performing this type of research to improve the validity, generalizability, and comprehensiveness of future empirical research in this area.

3.1 Case Study Context

For some ten years requirements from stakeholders in the transport logistics domain participating in European projects such as [1][2][3][4], have been captured and processed into the CF. The CF and its inherent semantic architecture specifies how information about transport logistic processes, information models and communication protocols can be applied to develop information systems for the transport logistics domain. The CF has been used in different projects in some specific activities: developing Transport Management Systems (TMS) components, building ontologies to be used for advanced logistics operations, eliciting functional requirements for the software development, driving the implementation of integration with legacy systems, etc. These activities are also used to further develop the content of the framework. The CF is known in Norway as ARKTRANS and in the European context as Common Framework. The CF includes different viewpoints that describe different parts of the transport logistics sector, involving the following architectural components (see Figure 2):

- A Reference Model decomposing the entire transport logistics sector into manageable sub-domains each addressing responsibility areas:
 - Logistics Demand: where demand for transport logistics services originates and where such services are being purchased.
 - Logistics Supply: domain responding to the needs of the Logistics Services Clients.
 - Transportation Network Management: ensures that the infrastructure is performing correctly, traffic flows properly, and that provides information about current and future capabilities of the transportation infrastructure.
 - Regulation Enforcement: The domain of those ensuring that all transport and logistics related activities are being performed according to rules and regulations.
- A set of Roles pertinent for each sub-domain, each role having its particular set of responsibilities within a sub-domain;
- A set of activities (UML Use Case models) and processes (BPMN models) diagrams defining relevant workflows for functions performed by the roles;
- A set of information models (UML Class models) describing relevant information flow between the sub-domains.

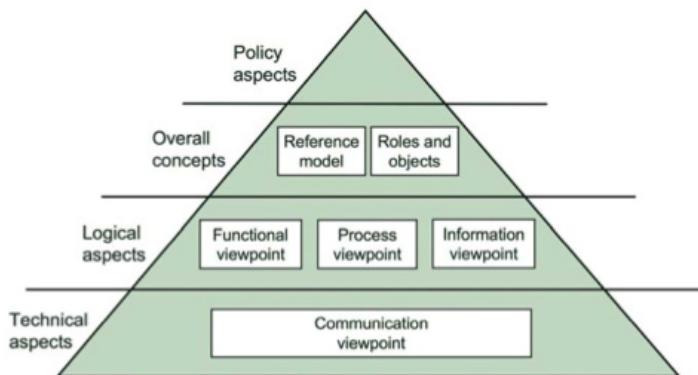


Fig. 2. Views on the CF

3.2 Case Study Design and Analysis

The main data collection methods were semi-structured interviews based on Maes and Poels framework [10][11] (See Appendix A). The difference in our approach is that we take a more exploratory and qualitative approach to the investigation, our approach aims to incentivize the users to talk about the issues on the CF. The first and second author of this paper conducted the interviews. The first author has large experience conducting interviews due to her background in empirical studies in software engineering. The second author is involved with the development of the framework in the last seven years. Each interview lasted approximately one hour, and the interviewees were informed about the audio recording and its importance to the study.

The subjects interviewed were different types of users of the framework (Table 2). We conducted interviews with seven users of the framework including three software developers, two requirements elicitors for transport logistics projects and two ontology builders. We also considered that the users of the framework had different experience profiles and used different parts of the CF, so when thinking about the improvements we knew if they were directed towards one specific part of the CF or in a general perspective. Also we asked them to answer their questions contextualizing on a specific case they would describe to us.

We used thematic analysis to analyze the data, a technique for identifying, analyzing, and reporting standards (or themes) found in qualitative data [19][20][21]. It is a way to recognize patterns in textual data, where emerging themes become categories for analysis. To support the data analysis, we used a tool, NVivo 9 [24], which enables information classification into searchable codes. Thematic analysis has limited interpretative power beyond mere description if it is not used within an existing conceptual framework [22]. We thus adopted the evaluation framework of Maes and Poels (Section 2.2) to anchor our analytic claims. Evaluation frameworks are useful as supports to better delineate qualitative studies and provide some clarity and focus; they can also be used to drive further discussion around the results [22].

Table 2. Background of the Users of the CF

Interviewee	Main Backgr.	Exp. in Transport Domain	# of projects using the CF.	Prev. Exp. with Models	Used Parts of CF
Requirements 1 (R1)	Road Transport	> 20 years	> Five	No	ARKTRANS, Reference and Functional View, Roles and Objects.
Requirements 2 (R2)	Maritim Transport	> 15 years	> Five	No	ARKTRANS, Reference Functional and Information View, Roles and Objects.
Ontology 1 (O1)	ICT	Two years	One	Yes	Common Framework, Roles and Objects, Information View
Ontology 2 (O2)	ICT	Two years	One	Yes	Common Framework, Roles and Objects, Information View
Developer 1 (D1)	ICT	Three years	> Three	Yes	Common Framework, Reference and Information View, Roles and Objects.
Developer 2 (D2)	ICT	Three years	> Three	Yes	Common Framework, Reference Functional, Process and Information View, Roles and Objects.
Developer 3 (D3)	ICT	> 15 years	> Three	Yes	ARKTRANS, Reference and Functional View, Roles and Objects.

To perform and thoroughly describe the thematic analysis, we used the recommended steps for performing thematic synthesis in SE that summarize the main themes constituting a piece of text [21]. The first step after the interviews was to transcribe all the interviews and then perform the “coding” of the material, performed by dissecting the text into manageable and meaningful text segments using a coding scheme according to the Maes and Poels evaluation framework, according to [20][22]. At this stage, 147 codes were generated. We then discussed each code in the data collection tool (NVivo 9). After the code generation, we reviewed each code in the raw information nature context. This step involved going through the text segments (extracts of the transcribed interview) in each code (or group of related codes) and extracting the salient, common, or significant themes in the coded text segments, so we could create the MindMap of the main important topics to discuss in the results. We returned to the original text, reading it linearly, theme by theme. The goal was to describe and explore the network, supporting the description with text segments. All authors of this paper discussed the rationale behind each theme. The objective here is to summarize the major themes that emerged in the network description and make explicit the patterns emerging in the exploration.

4 Case Study Results

Figure 3 shows the Mindmap generated from the results of the coding process performed on the transcripts of the interviews as described in the previous section.

Perceived Ease of Understanding (PEOU). On the perceived ease of understanding the results were divided in four main categories.

On the ease of understanding the interviewees stated that the model is understandable but it takes time. The reasons for these statements are that “there is too much information in the model and that it is difficult to get the whole picture”. One interviewee said: “of course at first sight you don’t have the complete model in your mind, but all the information you need is in there, I knew that if I wanted to know more on something, in the end it would be somewhere”. Another reason for problems in understanding the CF is that the model does not have a clear integration between the views and it is complex to try to map the views as described in Figure 1.

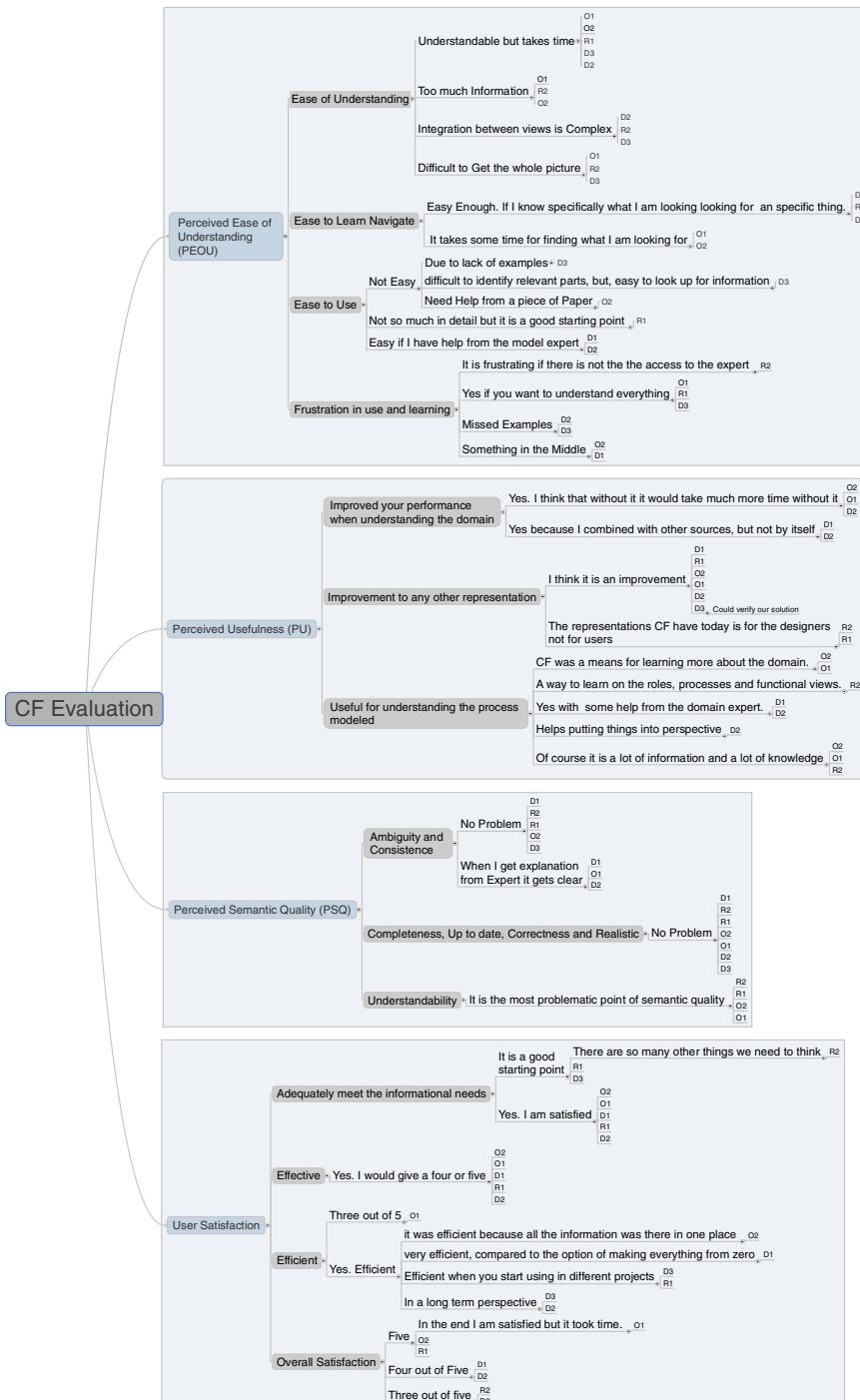
On the ease to learn and navigate aspect the interviewees were divided in two opinions, three interviewees thinks that easy enough to navigate the model, if they know specifically what they are looking for, and two thinks that it takes sometime to find what they are looking for. The reasons we believe there was divergence on the opinions, was the level of involvement with the model; the first three interviewees have been more involved with the CF, they have also been part of developing some parts of the CF, so they have outperformed the navigation barriers.

On the ease of use, there were again different opinions. One interviewee said that “it was a good starting point for eliciting the requirements” and two others stated that only when they had the help from the expert in the CF it was easier to use and another one said that when he had a piece of paper to help organizing the ideas it was possible to use. One interviewee said that it was not very easy to use in the detail but that it was a good starting point to the requirements specification. But for one developer it was not easy to identify the relevant parts.

On the frustration aspect, the interviewees mentioned again some of the same points they have mentioned before, that it is frustrating if there is not the access to the expert in the CF and also if they want to understand everything. One interviewee said: “...if you want to understand everything, it is a lot of work. A lot of things that you need to grasp, but if you have a clear goal, it is not too bad. I have been seen other models and compared to others, there is really no frustration on using the CF.”

The main points related to PEOU are that the model has too much information, so it takes time to understand and find the information needed and that also it requires the expert in the model to help on the understanding of the model.

Perceived Usefulness (PU). On the PU the interviewees stated that the use of the framework improved their performance when understanding the domain for the interviewees that were not expert in the domain before being introduced to the CF, their main background is ICT. Two developers said that it improved they performance only because they combined the use of the CF with talks with domain experts and participation in some meetings about the domain. The interviewees in general also believe that having the models is an improvement to other representation of the domain, but two of the interviewees thinks that the representations are too much oriented for designers of systems and not for the decision makers, so as a communication tool to the users it is lacking a view that shows the relevant parts of the framework. One interviewee stated that the models supported a common understanding among developers and integration between systems.

**Fig. 3.** - MindMap of the Results

Perceived Semantic Quality (PSQ). On the perceived semantic quality, as shown in the Mindmap. The CF is shown to be perceived as complete and up-to-date. The interviewees back up their answers based on the fact that the CF has been developed in the context of different European projects, these that the interviewees are mostly involved with. They also have not experienced much problems related to the correctness and realistic aspects of it, only one interviewee said that he has found some few mismatches and he will report back to the developers of the model.

Ambiguity and consistency did not show up to be a main problem as well, but three interviewees mentioned that when they have some problems understanding some part of the model, because of some ambiguity they contact the CF expert and get things sorted out. The only aspect that the interviewees mentioned to be the most problematic is the understandability of the model as already seen in the perceived ease of understanding subsection.

User Satisfaction (US). On the user satisfaction, on the overall the interviewees are satisfied with the model and how it meets their informational needs, two interviewees gave an overall score of three out of five, one said that “some things are good and some things are bad, I miss some other views of the model and a better harmonization with other standards”. One interviewee said: “in the end I am very satisfied with the CF but it took time”. On the effectiveness on using the CF, the users showed to be satisfied, and the overall perception is that the model is efficient when thinking in a long term perspective and after using the CF for different projects. They all also would recommend the use of the framework to other users, but we have some doubts on how honest they were in this question.

Advantages, Challenges and Improvements. One of the important highlights of the analysis was to identify in the answers the main perceived advantages, challenges and improvements to be made on the CF. We used all the answers in the questionnaire in addition to the specific questions to these aspects.

The identified advantages of the CF were that it was a very good starting point for the requirements specification of the IS transport systems, also that the CF provide enough information for their activities and helps to be consistent among different projects that they have used the framework for. The interviewees also mentioned that the framework is well documented and that it is a good starting point for learning about the domain, specially for the interviewees that have their background in ICT and that started to learn about the domain when they were involved with the usage of the CF. One important point also was that one of the interviewees said that the evolution of the framework is not problematic once that it is easy to contact the maintainers of the framework and get extensions or fixes are easily implemented.

The identified challenges, as mentioned before, the interviewees in general perceive the model as having too much information and too many levels, so it gets complex when trying to discuss the model in detail and also it is hard to get a broad picture of the model. It was clear also that there is a strong need of the expert in the model to guarantee the usability of the CF and that to be able to use the CF properly the user has to invest some time on it. And it takes time to get up to speed with the CF to be able to use it, one interviewee said that there is a “threshold time” that he is not sure how long it is, but it is there.

On improvements we identify the following action points:

- Create a better higher level view of the model; with a viewpoint that is closer to the decision makers and final users, showing examples of use and success stories of the use of the CF in real projects;
- Improve the navigation between the different viewpoints of the CF (Figure 2);
- Materialize the models with some concrete examples of the scenarios;
- Create mechanisms for keeping the users of the CF updated on the framework development and what will be the next improvements on the framework;
- Keep track of the changes so when there is a new version of the model, then the users can trace the changes and understand how the changes to the framework impact their use of it;
- Create training courses on the framework so the users can have a more unified view and understanding of the framework.

5 Limitations

The main limitations of our study were caused by the nature of the study and by our data analyses. First, qualitative findings are highly context- and case-dependent [27]. Three kinds of sampling limitations typically arise in qualitative research designs: cases that are sampled for observation (because it is rarely possible to observe all situations); time periods during which observations took place (problems of temporal sampling); and selectivity in the people who were sampled either for observations or interviews or in document sampling.

One possible threat to the trustworthiness of our data analysis is that we did not perform document analysis of the product of the activities performed by the users when using the framework, the study results if based on the users perceptions of the usage of the CF. Besides, We acknowledge that our involvement with the study may have hindered tougher criticisms.

Another possible threat to the trustworthiness of our data analysis is that the benefits of the strategy may have varied depending on the project, we did not account for the differences between projects, but to an overall perspective of the perceived views towards the CF. One way to overcome this limitation is that we interviewed people that are using the CF for many years and were involved in different projects during more than seven years. Besides, differences among subjects, the analysis was done considering that the interviewees had different backgrounds and domain knowledge.

On the transferability/generalizability of the results of this study to other settings or groups: to promote transferability, we described the selection and characteristics of the case study and we focused a section on describing the context and settings, data extraction, and synthesis process in detail, as well as quotations with our major findings. The results can then be used to compare the results we got with the results from other conceptual frameworks evaluations.

6 Conclusions

In this paper we have evaluated the quality of a conceptual model that is used in different European and Norwegian research projects. From a research point of view the

use of the evaluation model proposed by Maes and Poels, served to increase the number of studies investigating the mechanisms that leads to successful CMs. Conceptual model quality frameworks need to be tested empirically [5], specially, there is a need for evaluation of CMs that are used in practice and that are not only made for artificial testing of models notations. When having choosing such less artificial conceptual model, there is then the limitation on the number of users that have used the framework to perform some real activity related to the development of a IS using it. Therefore, this study was performed with a small but valuable number of users. On a researcher point of view, this study presents a qualitative way to study the quality of the CF model based on the user's perceptions.

The qualitative study based on interviews give the advantage that it provides a richer, detailed picture to be built up about why people act in certain ways towards the model, and their perceptions about the usage of the CF when compared to only ask the users which grade they would give to a certain aspect. Still we believe that the evaluation model can be improved to gather more details and deeper insights about some important quality issues on the model.

The study provided insights of what should be improved in the model, new ways to explore and to improve it. The study also provides us a way to build up a reliable evidence base to inform practice about the framework and made us aware of how the users really perceive the model. The degree to which conceptual models facilitate effective communication between analysts and key stakeholders is one important factor in influencing the continued use of conceptual modeling in organizations and we found out that there are some points of improvement is this direction, especially on providing higher level views and highlighting the examples of use of the CF. We have also found out that the problems in the model are stronger when related with the understandability of the model, but once that we were using a evaluation framework that was not focused on the understandability of the model, we could not explore and find out the specific reasons for the problems in the understandability of the model. The study was also important to us to be more aware of the importance of governance of the CF for the successful usage and sustainability of the model. We have also found out that the process view is not much used by the users, and this also may be related to issues related to the understandability of the models described in this view.

Future research will further explore the issues found in this paper, specially with the understandability issues. Furthermore, we will explore the quality dimensions and quality cornerstones as defined by Nelson et al. [23]. They have not yet developed specific metrics on each of the quality dimensions and that impeded us to completely use their framework in this study. Once defined, these metrics will be useful to both researchers and to practitioners to assure the quality of conceptual representations and the development of sound models and languages.

Acknowledgments. We would like to acknowledge the Research Council of Norway who funded the initial establishment of ARKTRANS and the MultiRIT, META and NonStop projects which developed ARKTRANS further and enabled the evaluation addressed by this paper; the European Commission who through the Freightwise, e-Freight, SMARTFREIGHT and iCargo projects funded the establishment of the Common Framework and this evaluation; the projects mentioned above; and users of conceptual models who provided input to this evaluation.

References

1. Natvig, M.K., et al.: ARKTRANS - The multimodal ITS framework architecture v6 (2007)
2. Freightwise Project, <http://www.freightwise.info/cms/>
3. e-Freight Project, <http://www.efreightproject.eu/>
4. SMARTFREIGHT Project, <http://www.smartfreight.info/>
5. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engin.* 55, 243–276 (2005)
6. Mendling, J.: Empirical studies in process model verification. In: Jensen, K., van der Aalst, W.M.P. (eds.) *Transactions on Petri Nets and Other Models of Concurrency II*. LNCS, vol. 5460, pp. 208–224. Springer, Heidelberg (2009)
7. Krogstie, J., Lindland, O.I., Sindre, G.: Defining quality aspects for conceptual models. In: *Proceedings of the 3rd IFIP 8.1 Working Conference on Information Systems*, Marburg, Germany, pp. 216–231 (1995)
8. Krogstie, J., Sindre, G., Jørgensen, H.D.: Process models representing knowledge for action: a revised quality framework. *EJIS* 15(1), 91–102 (2006)
9. Lindland, O.I., Sindre, G., Sølvberg, A.: “Understanding Quality in Conceptual Modeling. *IEEE Software* 11(2), 42–49 (1994)
10. Maes, A., Poels, G.: Evaluating quality of conceptual modelling scripts based on user perceptions. *Data and Knowledge Engineering* 63, 701–724 (2007)
11. Maes, A., Poels, G.: Evaluating Quality of Conceptual Models Based on User Perceptions. In: Embley, D.W., Olivé, A., Ram, S. (eds.) *ER 2006*. LNCS, vol. 4215, pp. 54–67. Springer, Heidelberg (2006)
12. Burton-Jones, A., Weber, R.: Understanding Relationships with Attributes in Entity-Relationship Diagrams. In: *20th Int. Conference on Information Systems*, pp. 214–228 (1999)
13. Gemino, A., Wand, Y.: Foundations for Empirical Comparisons of Conceptual Modeling Techniques. In: Batra, D., Parsons, J., Ramesh, E. (eds.) *Proc. of the Second Annual Symposium on Research in Systems Analysis and Design*, Miami, Florida (2003)
14. Gemino, A., Wand, Y.: A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering Journal* 9, 248–260 (2004)
15. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User acceptance of computer technology: A comparison of two theoretical models, *Manag. Science* 35(8), 982–1003 (1989)
16. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: toward a unified view. *MIS Quarterly* 27(3), 425–478 (2003)
17. Passos, C., Cruzes, D.S., Mendonca, M.: Beliefs Underlying Teams Intention and Practice: An Application of the Theory of Planned Behavior. In: *ESELAW 2013*, Uruguay (2013)
18. Passos, C., Cruzes, D.S.: Applying the theory of reasoned action in the context of software development practices: insights into teams intentions and behavior. In: *EASE 2013*, Brazil (2013)
19. Boyatzis, R.E.: *Transforming qualitative information: thematic analysis and code development*. Sage Publications (1998)
20. Braun, V., Clarke, V.: Using thematic analysis in psychology. *Qualitative Research in Psychology* 3(2), 77–101 (2006)
21. Cruzes, D.S., Dybå, T.: Recommended steps for thematic synthesis in software engineering. In: *ESEM 2011*, pp. 275–284. IEEE (2011)
22. Miles, M.B., Huberman, A.M.: *Qualitative Data Analysis: An Expanded Sourcebook*, vol. 2. Sage (1994)

23. Nelson, H.J., Poels, G., Genero, M., Piattini, M.: A conceptual modeling quality framework. *Software Quality Journal* 20(1), 201–228 (2012)
24. NVivo, QSR international (2010),
http://www.qsrinternational.com/products_nvivo.aspx
25. Fettke, P., Loos, P., Zwicker, J.: Business Process Reference Models: Survey and Classification. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 469–483. Springer, Heidelberg (2006)
26. Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., Bone, M.: The Concept of Reference Architectures. *Systems Engineering* 13(1), 14–27 (2010)
27. Patton, M.Q.: Enhancing the quality and credibility of qualitative analysis! *Health Services Research* 34 (5, pt. 2), 1189–1208 (1999)
28. Wand, Y., Weber, R.: An ontological model of an information system. *IEEE Transactions on Software Engineering* 16(11), 1282–1292 (1990a)

Appendix: Interview Questionnaire Guide

Background Information

- Education (Degree, Year). And, briefly, can you describe your previous knowledge on ICT and related to transport and logistics in general?
- Did you get any specific training in the CF (common framework)?
- Do you have previous experience with conceptual models as the common framework (CF)? Have you used the CF before? How many times?
- Have you used conceptual models before? If yes: How many times? In which activities?

About the process of using the CF

- Did you have other sources of information other than the CF?
- Did you use a specific method/tool?
- Did you perform an evaluation of the final product?

Evaluating the Usage of the CF. Based on one specific project you used the CF, please answer the following questions.

Perceived Ease of Understanding (PEOU)

- Was it easy for you to understand what the CF was trying to model? Can you talk about the process of trying to understand the framework?
- Was the use/learning of the CF frustrating? If yes: which parts? How did you deal with it?
- Was the CF easy to use? What do you think makes it easy (or not easy) to use?
- Was it easy to learn/navigate how to read the CF?

Perceived Usefulness (PU)

- Do you think that the CF was an improvement to a textual description of the domain knowledge? Any comments on that?
- Do you think that the CF was useful for understanding the process modeled? Or the relevant parts of the logistics domain? Why?
- Do you think that the CF improved your performance when understanding the logistics domain? Can you talk about it?

Perceived Semantic Quality (PSQ)

- Using the table. From 0 to 5. If you would grade the CF in terms of how the CF represents the domain of transport and logistics, in these categories which grade would you give?
- Comments on your answers? Can you provide examples?

Correctness	0	1	2	3	4	5
Ambiguity	0	1	2	3	4	5
Completeness	0	1	2	3	4	5
Realistic	0	1	2	3	4	5
Understandability	0	1	2	3	4	5
Consistency	0	1	2	3	4	5
Uptodate	0	1	2	3	4	5

User Satisfaction (US)

- From 0 to 5. Using the table. How satisfied are you with the CF for building the ontology? Any comments on your answers?

Other Comments:

- Any other comments? Would you like to talk about other advantages of using the CF? Or about the challenges of using the conceptual framework? Or what you think should be improved?

Adequately met my informational needs	0	1	2	3	4	5
Was efficient (maximum productivity and minimum wasted effort) in providing for my information	0	1	2	3	4	5
Was effective in providing for my information needs	0	1	2	3	4	5
I am satisfied with the CF for providing the information I needed	0	1	2	3	4	5

Towards the Effective Use of Traceability in Model-Driven Engineering Projects

Iván Santiago, Juan Manuel Vara,
María Valeria de Castro, and Esperanza Marcos

Kybele Research Group, Rey Juan Carlos University,
Avda. Tulipán S/N, 28933 Móstoles, Madrid, Spain

{ivan.santiago, juanmanuel.vara, valeria.decastro, esperanza.marcos}@urjc.es
<http://www.kybele.es>

Abstract. The key role of models in any Model-Driven Engineering (MDE) process provides a new landscape for dealing with traceability. In the context of MDE traces are merely links between the elements of the different models handled along the software development cycle. Traces can be consequently stored in models that can be processed by means of model-based techniques. To take advantage of this scenario, this paper introduces iTrace, a framework for the management and analysis of traceability information in MDE projects. We present the methodological proposal bundled in the framework as well as the tooling that supports it. Finally, a case study is used to introduce some of the functionalities offered by the framework.

Keywords: Model-Driven Engineering, Traceability, Trace Models.

1 Introduction

Despite the acknowledged significance that traceability should deserve in any Software Engineering activity [1], it is frequently ignored due to the inherent complexity of maintaining links among software artifacts [2]. The advent of Model-Driven Engineering (MDE, [3]) can drastically change this landscape. In particular, the key role of models can positively influence the management of traceability information since the traces to maintain might be simply the links between the elements of the different models handled along the process. Furthermore, such traces can be collected in other models to process them using any model processing technique, such as model transformation, model matching or model merging [4].

To confirm this assumption, in previous works [5] we have conducted a systematic review to assess the state of the art on traceability management in the context of MDE. One of the main conclusions derived from such review was that the operations most commonly addressed by existing proposals are storage, visualization and CRUD (creation, retrieval, updating and deletion) of traces. By contrast, those less commonly addressed are the exchange and analysis of traceability information. In particular, the analysis of traceability information was addressed only by 32.35% of the proposals analyzed.

To deal with the lack of proposals focused on the analysis of traceability information, this work introduces **iTrace**, a methodological and technical framework for the management of traceability information in MDE scenarios. As will be shown later, **iTrace** automates the production of trace models and supports different types of analysis, at different levels of abstraction, of the raw data provided by such traces.

The rest of this paper is structured as follows: Section 2 reviews related works in the area; Section 3 presents the process for the production of trace models; Section 4 illustrates the proposal by means of a case study; and finally, Section 5 summarizes the main conclusions derived from this work and provide some directions for further work.

2 Related Works

There are several works dealing with traceability in the context of MDE. Most of them deal with storage, visualization, semi-automatic generation, and CRUD operations for traces but only a few of them are focused on the analysis of traceability information. Those dealing with analysis can be grouped into three big categories attending to their main focus: 1) impact analysis [6] [7] [8]; 2) generation of traceability reports [9] [10] and 3) identification and classification of traces [11].

Regarding the works focused on impact analysis, the works from Anquetil *et al.* [6] and Walderhaug *et al.* [8] present proposals to extract information from trace links repositories in order to identify all the artifacts potentially impacted by a change. The former accompanies the proposal with a supporting tool. Likewise, Olsen and Oldevik in [7] presented a prototype where linked artifacts are displayed when source model elements are selected.

Next, we consider two different tools focused on the generation of traceability reports: **TraceTool** and Safety Requirements Trace Tool(SRTT). The former was presented by Valderas and Pelechano in [9]. It generates HTML traceability reports describing how the elements of a navigational model have been derived from a requirements model. The latter was presented in [10] by Sanchez *et al.* and consumes trace models to generate HTML traceability reports in the context of model-driven development of tele-operated services robots. Both tools are focused on requirements traceability, i.e. they do not consider low-level traceability, such as those produced when moving from design models to working-code.

Finally, in [11] Paige *et al.* present the Traceability Elicitation and Analysis Process (TEAP). TEAP elicits and analyzes traceability relationships in order to determine how they fit into an existing traceability classification. Classifying trace links can help to understand and manage them. Although this proposal is very useful for MDE practitioners, it can be difficult to understand for a business analyst or an end-user.

Regarding the current state of the art, **iTrace** provides a number of contributions: first, it automates the production of traces in MDE projects. To that end, it bundles different components to inject traceability capabilities into existing

model transformations. Next, it considers all the models involved in the development process and consequently it does not limit its scope to requirements traceability. In addition, it provides new editors for trace models edition that fit better with the relational nature of trace models. Besides, all the works reviewed, except the one from Walderhaug *et al.* [8], supports ad-hoc analysis. That is to say, the analyzing process proposed fits in the particular MDE projects considered by each proposal. By contrast, **iTrace** aims at providing a generic proposal that can be applied to any given MDE project. Finally, one of the most relevant features of **iTrace** regarding existing proposals is the fact that it supports multidimensional analysis from trace links. This feature will be later introduced in this paper.

3 Supporting the Production and Analysis of Trace Models

This section introduces our proposal to support the analysis of traceability information by outlining the main steps of the process and presenting the technical components that comprise the **iTrace** framework¹.

The starting point of the analysis process supported by **iTrace** is an existing MDE project. Note that we aim at defining a "project-agnostic" proposal, i.e. a proposal that can be applied to any type of project. Thus, to illustrate the idea we consider as starting point a generic MDE project, i.e. one that includes models at different abstraction levels (including weaving models) plus a set of model transformations connecting them (both m2m and m2t). This way, left-hand side of Fig. 1 shows a simple project, composed of three models, namely **Ma**, **Mb** and **Mc** and two m2m transformations connecting them (**Ma2Mb** and **Mb2Mc**).

The rest of Fig. 1 provides a tabular overview of the **iTrace** framework: the **iTrace Component** row shows the technical components supporting each step of the process, while the **Process** row shows the artifacts (models, transformations, etc.) handled. Finally, the **Notation** row shows a brief legend to ease the understanding of the different graphic elements.

During the **Identification** stage, an existing project composed of models and transformations is selected as input. In the **Addition** phase, the transformations are enriched by means of Higher-Order Transformations (HOTs) [12] following the idea first-sketched by Jouault in [13]. The enrichment process bundled in **iTrace** is a little bit more complex than the one from [13], due to the increased complexity of **iTrace** metamodels.

Next, the enriched transformations are run during the **Execution** phase in order to generate the corresponding target model/s plus the trace models that connect their elements. During the **Union** phase, another transformation consolidates such trace models into a unique **iTrace** model.

In some sense, the trace models generated at this point are *normalized* models. However, we have found *denormalized* trace models to be more convenient to

¹ Full screen images for all the figures can be found in

<http://www.kybele.etsii.urjc.es/er/>

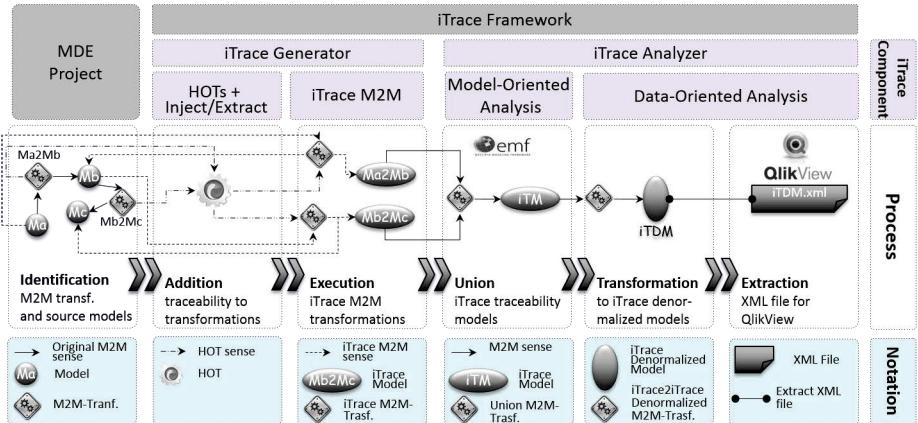


Fig. 1. iTrace process overview

run the queries that perform the Data-Driven Analysis supported by iTrace. This idea is brought from the databases area, where normalized data models are used for day-to-day activities (mainly CRUD operations), while denormalized models are preferred for read-intensive operations, like those supported by data-warehouses [14]. In some sense, the Data-Oriented Analysis of traces can be seen as the analysis of the historical data collected during the development of the project. These data are persisted in the shape of trace models. Therefore, during the **Transformation** phase, a m2m transformation consumes the normalized trace models generated in the **Union** phase to produce a denormalized trace model (iTrace denormalized model).

Finally, during the **Extraction** phase, the iTrace denormalized model is serialized into an XML file that is later imported into QlikView² in order to populate the different dashboards used to support the Data-Oriented Analysis provided by iTrace. As a result, high-level overviews of the relationships between the artifacts involved in the development process are obtained from the trace models produced in the project. Next section illustrates the idea by showing two particular scenarios of Data-Oriented Analysis by means of a case study.

4 Case Study

As previously mentioned, iTrace aims at improving the use of traceability information in MDE projects. This section shows how it is effectively done for an existing MDE project. To that end, we lean on M2DAT-DB [15], a framework for Model-Driven development of modern database schemas that support the whole development cycle, from PIM to working code. In particular, M2DAT-DB supports the generation of ORDB schemas for Oracle and the SQL:2003 standard as well as XML Schemas from a conceptual data model represented with a UML class

² QlikTech International AB. <http://www.qlikview.com>

diagram. In this work we focus on three of the model transformations bundled in M2DAT-DB: the UML2SQL2003 transformation produces a standard ORDB model from a UML class diagram; the SQL20032ORDB transformation generates an ORDB model for Oracle from the standard one; finally, the ORDB2SQL2003 transformation implements the inverse transformation. Besides, apart from the corresponding source models, each transformation is able to consume an annotation model to introduce some design decisions in the transformation process.

In the project under study such transformations are run to produce the Online Movie Database (OMDB) schema. The OMDB is a case study presented by Feuerlicht *et al.* [16] that has been also used by some other authors. Using an “external” case study prevents us from using ad-hoc models that might fit better to our purposes. The OMDB is devised to manage information about movies, actors, directors, playwrights and film-related information.

Next sections present two of the dashboards provided by *iTrace*. They are populated with the data collected in the *iTrace* models produced when M2DAT-DB is used to generate the OMDB schema. First dashboard can be used to get an overview of the transformations involved in the process. Second dashboard provides a deeper view of the mapping rules and their workload.

4.1 Mapping Rules Overview

Model transformations are inherently complex [17]. Therefore, dealing with legacy transformations might be even more complex. The analysis of trace models can be used in this context to raise the abstraction level at which they think about model transformations. In addition, it allows the developer to abstract from the particular model transformation language used and provide him with simple and comprehensible information regarding the mapping rules that compose the different transformations.

For instance, Fig. 2-a shows a code excerpt from the UML2SQL2003 transformation. In particular, it shows the `MemberEnd2NotNullOnTT` mapping rule. To understand the functionality of this rule and what type of elements it maps, a minimum knowledge of ATL is needed.

By contrast, a quick look at the dashboard beside shows at first sight which the purpose of the mapping rule is. Indeed, no previous knowledge of ATL is needed. The information displayed on the upper side of the dashboard reveals that the rule maps objects from UML models into SQL2003 objects. In particular, the lower side of the dashboard shows that the rule is responsible for mapping pairs of `Class` and `Property` objects into `NotNull` objects.

Note that the dashboard provides this type of information for all the transformations that conform the project under study. The upper side of the dashboard provides a set of filters that allow the user to select a particular transformation, source model, target model or mapping rule/s (non-selected filtering values are greyed-out). The bottom part of the dashboard shows the type of elements transformed by the mapping rules that meet the criteria established by the user. Note also that this analysis may be useful in order to document not only m2m transformations, but also m2t ones.

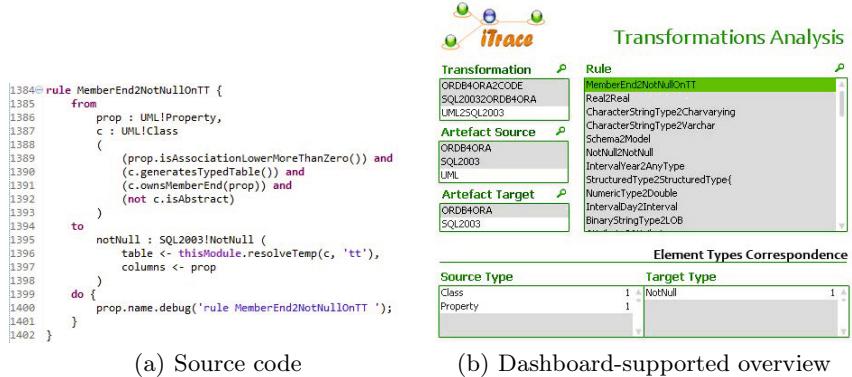


Fig. 2. Two ways of looking at the MemberEnd2NotNullOnTT mapping rule

4.2 Mapping Rules Workload

Previous section has shown how trace models can be processed to provide technology-independent overviews of the model transformations involved in a given project. The dashboard shown in Fig. 3 goes a step further, since it provides a closer look at the transformations. In particular, it allows identifying which are the rules involved in a given transformation and which is their workload of such rules, i.e. the amount of objects effectively mapped by the mapping rule under consideration.

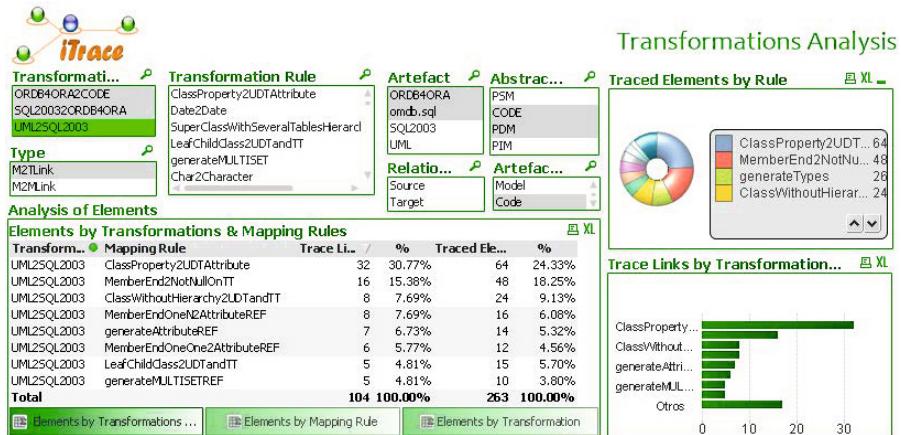


Fig. 3. Mapping rules workload dashboard

To that end, upper side of the dashboard bundles a set of controls to define high-level criteria for the analysis. This way, the traces that will be analyzed can be filtered according to:

- Transformations: only traces produced by the select model transformations will be considered.
- Type: only model-to-model or model-to-text traces will be considered.
- Transformation Rule: only traces produced by the selected mapping rules will be considered.

Likewise, the model elements that will be object of consideration can be filtered according to another set of criteria:

- Artefact: only elements included in the selected models or source-code files will be considered.
- Relation Type: depending on the selection, only model elements used that were either as source or target objects of the selected transformations will be considered.
- Abstraction Level: only model elements belonging to models defined at the selected abstraction levels will be considered.
- Artefact Type: depending on the selection, only model elements or source-code blocks will be considered.

Obviously, none of the criteria above are mandatory. That is to say, the user might set no values for any of them. If so, no filtering is applied and every trace (respectively model element) is considered in the analysis.

Once the criteria have been fixed (if any), the central and lower part of the dashboard collects aggregated data regarding the number of traces, model elements (referred to as **traced elements**) and source-code blocks, which fulfill the criteria. In this case, the table in the middle shows which are the mapping rules producing more traces. In particular, it shows the top 8 rules, while the rest are blended into the **Others** row.

First and the second columns show respectively the transformations and mapping rules rules under consideration (those that meet the filtering criteria). Next columns show the number of trace links produced and model elements referenced by each mapping rule and the percentage over the total number of traces produced by the mapping rules selected. For instance, second row of the table states that the **MemberEnd2NotNullOnTT** of the **UML2SQL2003** transformation generates 16 trace links (15.38% over the total number of trace links produced by the selected mapping rules) and such links refer to 48 model elements (note that not every trace link represents a one-to-one relationship).

Finally, the right side of the dashboard provides different views of these data. The pie chart represents the distributions of traced elements by transformation rule, while the bar graph summarizes the number of trace links produced by each transformation rule. As previously suggested, the information collected in this dashboard could be used by model-transformation developers to locate candidates for refining. For instance, the data presented in Fig. 3 highlights the importance of the **ClassProperty2UDTAttribute** mapping rule, which generates more than 30% of the trace links produced by the **UML2SQL2003** transformation. Thus, this rule might be object of study if transformation developers aims at optimizing the overall performance of the transformation.

5 Conclusion

This paper has introduced **iTrace**, a model-driven framework for the management of traceability information in MDE projects. The framework supports the production of trace models from any existing project and the subsequent analysis of such models. For instance, the dashboards from the case study have shown that the information provided by **iTrace** can be used to produce a high-level overview of the transformations involved in a project, to explain the purpose of a particular mapping rule or to identify the rules that should be optimized in order to improve the execution of a given transformation.

Two main lines are distinguished regarding directions for further work. On the one hand, we are extending the proposal to support other transformation languages in order to show that it can be effectively used with any metamodel-based language. Besides, we are integrating support for text-to-model transformations so that Model-Driven Reverse Engineering [3] projects can be also object of study.

Acknowledgments. This research is partially funded by the MASAI project, financed by the Spanish Ministry of Science and Technology (Ref. TIN2011-22617).

References

1. Asunción, H.U.: Towards practical software traceability. In: Companion of the 30th International Conference on Software Engineering, ICSE Companion 2008. ACM, New York (2008)
2. Oliveto, R.: Traceability Management meets Information Retrieval Methods - Strengths and Limitations. In: 12th European Conference on Software Maintenance and Reengineering, CSMR, pp. 302–305 (2008)
3. Schmidt, D.C.: Model-Driven Engineering. IEEE Computer 39, 25–31 (2006)
4. Bernstein, P.: Applying model management to classical meta data problems. In: First Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, pp. 1–10 (2003)
5. Santiago, I., Jiménez, A., Vara, J.M., De Castro, V., Bollati, V., Marcos, E.: Model-Driven Engineering As a New Landscape For Traceability Management: A Systematic Review. Information and Software Technology 54, 1340–1356 (2012)
6. Anquetil, N., Kulesza, U., Mitschke, R., Moreira, A., Royer, J., Rummler, A., Sousa, A.: A model-driven traceability framework for software product lines. Software and Systems Modeling 9, 427–451 (2010)
7. Olsen, G.K., Oldevik, J.: Scenarios of traceability in model to text transformations. In: Akehurst, D.H., Vogel, R., Paige, R.F. (eds.) ECMDA-FA. LNCS, vol. 4530, pp. 144–156. Springer, Heidelberg (2007)
8. Walderhaug, S., Johansen, U., Stav, E., Aagedal, J.: Towards a generic solution for traceability in mdd. In: European Conference on Model-Driven Architecture - Traceability Workshop (ECMDA-TW 2006), Bilbao, Spain, pp. 41–50 (2006)
9. Valderas, P., Pelechano, V.: Introducing requirements traceability support in model-driven development of web applications. Information and Software Technology 51, 749–768 (2009)

10. Sánchez, P., Alonso, D., Rosique, F., Álvarez, B., Pastor, A., Pastor, J.A.: Introducing Safety Requirements Traceability Support in Model-Driven Development of Robotic Applications. *IEEE Transactions on Computers* 60, 1059–1071 (2011)
11. Paige, R., Olsen, G., Kolovos, D., Zschaler, S., Power, C.: Building model-driven engineering traceability classifications. In: Proceedings of the 4th European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW 2008), Berlin, Germany, pp. 49–58 (2008)
12. Tisi, M., Cabot, J., Jouault, F.: Improving higher-order transformations support in ATL. In: Tratt, L., Gogolla, M. (eds.) ICMT 2010. LNCS, vol. 6142, pp. 215–229. Springer, Heidelberg (2010)
13. Jouault, F.: Loosely coupled traceability for ATL. In: Proceedings of the European Conference on Model Driven Architecture (ECMDA) Workshop on Traceability, Nuremberg, Germany, vol. 91 (2005)
14. Sanders, G.L., Shin, S.: Denormalization Effects on Performance of RDBMS. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences, vol. 3. IEEE Computer Society, Washington, DC (2001)
15. Vara, J.M., Marcos, E.: A framework for model-driven development of information systems: Technical decisions and lessons learned. *Journal of Systems and Software* 85, 2368–2384 (2012)
16. Feuerlicht, G., Pokorny, J., Richta, K.: Object-Relational Database Design: Can Your Application Benefit from SQL:2003? In: Information Systems Development, pp. 975–987. Springer (2009)
17. Bollati, V.A., Vara, J.M., Jiménez, Á., Marcos, E.: Applying MDE to the (semi-)automatic development of model transformations. *Information and Software Technology* 55, 699–718 (2013)

Modeling Citizen-Centric Services in Smart Cities

Sandeep Purao¹, Teo Chin Seng², and Alfred Wu²

¹ College of IST, Penn State University, University Park, PA, USA

² iCity Laboratory, Singapore Management University, Singapore

spurao@ist.psu.edu

Abstract. Much research about large cities has focused on *policy-level* action for concerns such as infrastructure, basic amenities and education, treating the citizens as a collective of millions. We suggest an alternative, drawing on recent moves towards ‘digital era governance.’ We argue for and develop a foundation for the design and modeling of services that focus on *individuals*. Drawing on and extending prior work in service modeling and public-sector governance, we develop a formalism for modeling citizen-centric services, illustrate its application, and extract principles that underlie this effort. The paper concludes with pointers to other aspects of the larger iCity project, aimed at building smart cities in the world’s rapidly growing regions.

Keywords: Mega-cities, Smart city, Citizen-centric services, Service modeling, Case management, Digital era governance, iCity.

1 Introduction

For the first time in human history, more of us live in urban, not rural areas (Dugger 2007). In the near future, this surge is expected to produce many more mega-cities (with populations of 10 million+). By 2050, urban centers are expected to house 2 out of every 3 individuals on the planet (UN 2012). This change will be particularly extraordinary in Asia where “the accumulated urban growth ... during the whole span of history will be duplicated in a single generation” (Dugger 2007). The challenges facing these fast-rising Asian cities are, thus, qualitatively different from those for existing cities (Woetzel et al. 2009).

Challenges that need to be addressed include infrastructure concerns such as transportation (World Bank 2013), provision of utilities such as water (Dobbs et al. 2012), and governance for services such as health and education (WHO 2008; UNESCO 2012). Each has engendered *policy-level* action. Advances in research are, however, pointing out that it is important to create solutions for each *individual* citizen, not just the collective of millions. Work in public administration research has moved past the focus on so-called new public management (with a focus on markets and efficiencies (see Ferlie et al. 1996)) to a focus on services targeted at individuals. This new wave, described as digital era governance or DEG (Dunleavy et al. 2006; Dunleavy et al 2008) provides the inspiration for the work described in this paper.

The *objective* of this research is to develop and demonstrate a meta-model for the design and modeling of citizen-centric services in smart cities. The key motivation

and precursors to our work are outlined above. The paper begins with a scenario in a narrative style that describes Sam, a citizen with a chronic disease, who provides the focal point for development of the meta-model, which recognizes the interaction between the *specific* individual context and *generic* service offerings from government agencies. The paper uses multiple episodes not only to highlight key constructs in the meta-model but also to surface fundamental principles that can drive the design of such services (Dunleavy et al 2010, Fishenden and Thompson 2012). We acknowledge recent work by Bergholtz et al (2011) and Andersson et al (2012), which provides important precursors to our work. The key *contributions* of the research, therefore, are a formalism that provides the foundation for translation of government programs to citizen-centric services in smart cities.

2 Background and Prior Work

2.1 City Governance: Rethinking the Role of IT and Services

IT-enabled delivery of public services in cities is not new. The last decades of the 20th century have witnessed significant injection of IT in support of public services (Fishenden and Thompson 2012; Gil-Garcia and Martinez-Moyano, 2005). The first generation mirrored advances in the private sector to increase automation and reduce costs (Cordella and Iannacci 2010). Here, IT applications were seen as tools to minimize errors and get things done faster. The second generation (described as new public management or NPM) witnessed the adoption of market-based mechanisms to drive greater efficiencies (Fang 2002; McNulty and Ferlie 2004). IT platforms were seen as facilitators for public-private competition but failed to deliver the anticipated benefits; and instead, lead to administrative complexity as well as difficulties in coordinating service delivery. IT, thus, wrought fossilization of silos making the delivery of citizen-centric service an impossibly difficult outcome to achieve.

The response, the third generation, is characterized by open standards and architectures. Here, IT platforms are seen as enablers of outcomes, made possible by separation of service logic from the supporting applications (see Fishenden and Thompson 2012). Described by Dunleavy et al. (2006) as Digital Era Governance (DEG), this generation aims for “re-aggregation of public services under direct government control around the citizen,” made possible by dis-aggregation of previously grouped functionalities for extraction and publication of services, which then might be personalized and re-aggregated. This juxtaposition - between the policy-level efforts and the unique needs of individuals - is at the core of new thinking about IT in the public sector, and provides the inspiration of our work.

2.2 Service Design and Modeling

The notion of services and service design (Goldstein et al. 2002) is at the heart of efforts needed to realize the vision of digital era governance (DEG). Scholars in several disciplines have addressed this area of work. Examples include work related to service-dominant logic (Lusch and Vargo 2006), customer service design

(Segelström 2010), the design of web services (Papazoglou and Yang 2002), exploration of service-product design (Giannini et al. 2002) and exploration of process and use views (Andersson et al. 2012). As a result, the fundamental terminology has remained in flux. For the purpose of this research, we co-opt and extend several efforts (see Wikipedia 2013) to define a service as:

- the seeking of an outcome by a citizen,
- where the outcome has a quantifiable value,
- and achieving it requires multiple interactions across different agencies,
- drawing upon specific capabilities of these agencies
- in a manner that takes place over time and repeated as necessary

This definition emphasizes several properties including the co-construction of outcomes (Gebauer et al. 2010), suggesting specific actions that the recipient must take to realize the outcome of value; and the primacy of an individual case for the realization of benefits. Using these as starting points, we develop our proposal as a meta-model for the design and modeling of citizen-centric services.

3 A Foundation for Modeling Citizen-Centric Services

The development of the formalism is grounded in several scenarios that were obtained via discussions and interviews, and distilled through multiple revisions. Here, we begin with a condensed scenario that describes Sam, an individual dealing with a chronic disease.

Scenario: Sam, a 45-year old cab driver has been living alone. He has been diagnosed with type-2 diabetes. Following the diagnosis, he has been enrolled in a home care system to manage the chronic disease. A nurse-educator has taught him how to administer the three insulin injections a day and watch for onset of other problems. Sam has worked with a dietician for a personalized diet plan, and a physiotherapist for an exercise regimen. Sam's smart phone is able to access his history, prescriptions and plans. His calendar alerts him for administering medication and tracks his exercise and diet actions. A social worker helps Sam to connect with support groups and social activities. After a year, when Sam's condition deteriorates; he cannot perform his job as a cab driver. He is admitted to the hospital. The social worker helps Sam apply for financial assistance, respecting his privacy. Doctors at the hospital are able to access, with Sam's consent, not only his medical history but also other details of his case.

A number of observations can be made from the narrative. A City will have several agencies that provide the services Sam will need. Sam will consume these services over a period of time, as needed. The provision of services will require personalization, when the services are enacted during specific encounters. Many will need follow-ups from Sam. A holistic view of an individual case will then be a combination of service performance and actions from the individual. Figure 1 captures these concepts.

Agencies and Capabilities

$a_1..a_i \in A$	set of agencies
$c_1..c_j \in C$	set of capabilities
$\text{capable-of}(a_i, c_j)$	capability c_j is provided by agency a_i {0,1}
$\forall c_j \in C \exists a_i \in A \text{capable-of}(a_i, c_j) = 1$	a capability is provided by at least one agency

Individual Citizens and Agency Representatives

$d_1..d_k \in D$	set of citizens
$b_1..b_y \in B$	set of abilities
$\text{able-to}(d_k, b_y)$	ability b_y is available for citizen d_k {0,1}
$r_1..r_z \in R$	set of representatives
$\text{agent-of}(a_i, r_z)$	representative r_z represents agency a_i {0,1}
$\forall r_z \in R \exists a_i \in A \text{agent-of}(a_i, r_z) = 1$	a representative represents at least one agency

Service Offerings, Encounters, Actions

$s_1..s_l \in S$	set of potential service offerings
$s_l = \text{bundle-of}\{c_1..c_j\}$	service s_l is offered by bundling capabilities c_1 to c_j
$e_1..e_m \in E$	set of encounters
$e_m = \text{performance}(s_l, d_k, r_z, t)$	encounter as enactment of s_l for d_k by r_z at time t {0,1}
$g_1..g_n \in G$	set of actions (preceding or following an encounter)
$g_n = \text{enactment}(d_k, b_y, t)$	action as exercise of b_y by citizen d_k at time t {0,1}
$\text{precedes}(e_1, e_2)$	encounter e_1 precedes action e_2
$\text{follows}(e_1, e_2)$	encounter e_1 follows action e_2
$\text{linked}(e_m, g_n)$	encounter e_m and action g_n are linked {0,1}

Episodes and Cases

$f_1..f_v \in F$	set of episodes
$f_v = \{<e_m..g_n>\} [t]$	episode as a set of encounters and actions in timespan t
$\text{episode-of}(f_v, d_k)$	episode f_v belongs to citizen d_k {0,1}
$h_1..h_w \in H$	set of cases
$h_w \in \{f_v\}$	case as an ordered set of episodes
$\text{belongs-to}(h_w, d_k)$	case h_w belongs to citizen d_k {0,1}
$\text{visible}(f_v, r_z)$	episode f_v is visible to representative r_z {0,1}
$\text{access}(h_w, r_z)$	case h_w is visible to representative r_z {0,1}
$\text{visible}(f_v, r_z) = 1 \forall f_v \in h_w \Leftrightarrow \text{access}(h_w, r_z) = 1$	visibility on all episodes in a case implies case access

Fig. 1.A Set of Formalisms for the Designing Citizen-centric Services

The formalism and the accompanying meta-model (not shown here) provide the foundation for defining service-offerings, customizing these for individual citizens, ensuring cross-agency coordination within the context of the individual, and creating and sustaining the longitudinal case for the citizen. It is important to draw parallels with and distinctions from the formalisms described above and the concepts developed by Bergholz et al (2011) and Andersson et al (2012). For example, ideas related to capability, service offering and specific instances they call event have parallels in our formalism respectively as: capability, service offering and action. Our work also introduces concepts such as encounter and case, similar to but not identical to the ideas of process in theirs. The differences are driven by our context (smart cities) compared to the context (business) implicit in the work by Andersson et al (2011). Although it is possible to engage in these comparisons further, we move to demonstrating application of the model in light of space constraints. Our intent in this effort is both, demonstration of the formalism as well as an initial evaluation of feasibility in the context of smart cities.

4 Application and Evaluation

The scenario earlier is a condensed version of a larger narrative obtained from agency representatives and individual citizens. To demonstrate application of the formalisms, we decomposed it in several episodes, of which one is elaborated here. Table 1 summarizes the episodes. The episode and the model appear after the table.

Table 1. Episodes derived from the scenario

Episode	Description	This Paper
Diagnosis	Sam's initial diagnosis after the onset of symptoms	
Home Care	Sam's enrollment into Home Care system with instructions	X
Self-care	Personalized services direct Sam towards diet and exercise	
Management	Sam's efforts to manage the chronic disease	
Escalation	Escalation of the disease and Sam's loss of job	
Case Analysis	Analyzing Sam's case for ongoing help	
Analytics	Analysis of multiple cases, including Sam's	

Episode: Home Care. Sam was enrolled in home care to manage his diseases. A summary of his case notes, clinical information and prescriptions were transferred to his smart phone. He was offered education sessions via prompts on his smart phone and email. Based on his driving schedule, Sam selected a session. During the first appointment, a nurse interviewed Sam on his lifestyle, diet preference, habits and addiction to measure his baselines. During his second appointment, the nurse trained him on the disease and how it can be managed. He now knew how to look for signs of complications, was taught insulin injection technique and made aware of symptoms so he may recognize the onset of problems like hypoglycemia or hyperglycemia. The nurse asked him for an emergency contact; Sam shared information about his brother.

Figure 2 shows the interaction diagram for Episode 2, keeping extraneous information to a minimum (based on our intent of demonstration and evaluation).

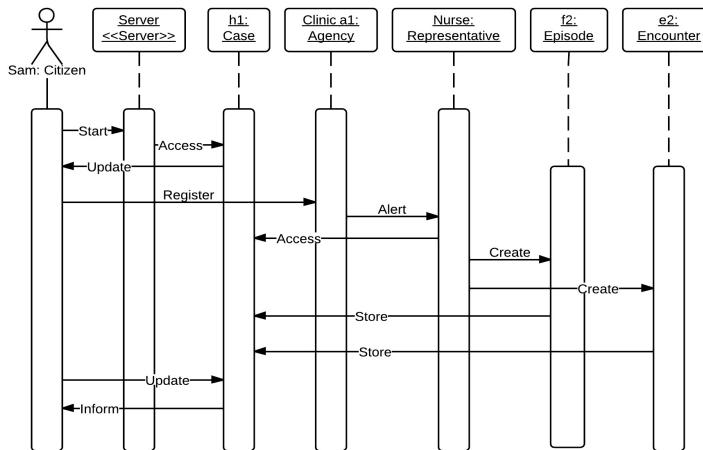


Fig. 2. Interaction Diagram for the Home Care Episode

The episode demonstrates how the formalism (see Figure 1) may be used to design and activate service offerings and encounters. The interactions (example shown in Figure 2) describe how the episodes can be operationalized. The set of interaction models (or all episodes) also provide a vehicle to discover principles that underlie the formalisms. Table 3 describes the outcomes of this investigation and reflection.

Table 2. Key Principles for Design of Citizen-centric Services

Principle	Description
Bridging Abstractions	Bridging from the abstraction of agency policies to specific service offerings for individuals
Aggregation Levels	Aggregation of different agency capabilities to different service offerings
Personalization of Services	Personalization of services to the needs of individual citizens at different times
Independent Action	Independent follow-up actions from citizens based on the services performed by the agency
Accumulation to Cases	Longitudinal accumulation of encounters and actions to episodes; Accumulation of episodes to cases

5 Discussion and Concluding Remarks

This research builds upon prior work related to digital era governance (Dunleavy et al. 2006; Fishenden and Thompson 2012) and service modeling and design (Bergholtz et al. 2011; Andersson et al. 2011). The key contribution of work is a formalism that can act as the foundation for the design and modeling of citizen-centric services in smart cities. Application to multiple episodes has allowed us the opportunity to uncover principles that can underlie service design and modeling for citizen-centric services. These are a core contribution of our work.

The key motivation for this work is the rise of mega-cities around the globe, and the resulting demands on ensuring digital era governance (DEG). Appropriate design and modeling of services is of primary importance in this context. The formalisms we have developed stand *in contrast* to the previous generations (such as NPM). Practical applications of our model are, therefore, direct and straightforward. We acknowledge that effective citizen-centric services will require more than a set of formalisms (see Peters and Savoie 1995). These concerns, such as coordination challenges and incentives for changing practices, remain concerns that need to be addressed.

Acknowledgements. We acknowledge support from the SMU-TCS iCity project, funded by Tata Consultancy Services; contributions of scenarios and detailed episodes from our project partners; and comments from anonymous reviewers.

References

1. Andersson, B., Bergholtz, M., Johannesson, P.: Resource, Process, and Use – Views on Service Modeling. In: Castano, S., Vassiliadis, P., Lakshmanan, L.V.S., Lee, M.L. (eds.) ER Workshops 2012. LNCS, vol. 7518, pp. 23–33. Springer, Heidelberg (2012)
2. Bergholtz, M., Johannesson, P., Andersson, B.: Towards a Model of Services Based on Co-creation, Abstraction and Restriction. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 476–485. Springer, Heidelberg (2011)
3. Cordella, A., Iannacci, F.: Information systems in the public sector: The e-Government enactment framework. *Journal of Strategic Information Systems* 19(1), 52–66 (2010)
4. Dobbs, R.: Urban world: Cities and the rise of the consuming class. *Mckinsey Global Institute. Insight Report* (June 2012)
5. Dugger, M.: Half the world's population will live in cities next year, June 27. *Times*, New York (2007)
6. Dunleavy, P., Margetts, H., Bastow, S., Tinkler, J.: New Public Management is dead. Long live digital-era governance. *Journal of Public Administration Research and Theory* 16(3), 467–494 (2005)
7. Dunleavy, P., Margetts, H., Bastow, S., Tinkler, J.: Digital Era Governance: IT Corporations, the State and e-Government. Oxford University Press, Oxford (2008) (revised edition)
8. Dunleavy, P., et al. (eds.): Understanding and Preventing Delivery Disasters in Public Services. Political Studies Association Conference, Edinburgh (2010)
9. Fang, Z.: E-Government in Digital Era: Concept, Practice, and Development. *International Journal of The Computer, The Internet and Management* 10(2), 1–2 (2002)
10. Ferlie, E., et al.: The new public management in action. Sage, London (1996)
11. Fishenden, J., Thompson, M.: Digital Government, Open Architecture, and Innovation: Why Public Sector IT Will Never Be the Same Again. *Journal of Public Administration Research and Theory* (2012) (forthcoming)
12. Gebauer, H., Johnson, M., Enquist, B.: Value co-creation as a determinant of success in public transport services: A study of the Swiss Federal Railway operator (SBB). *Managing Service Quality* 20(6), 511–530 (2010)
13. Giannini, F., et al.: A modelling tool for the management of product data in a co-design environment. *Computer-Aided Design* 34(14), 1063–1073 (2002)

14. Gil-Garcia, J.R., Martinez-Moyano, I.J.: Exploring E-Government Evolution: The Influence of Systems of Rules on Organizational Action, WP No. 05-001. National Center for Digital Government, Univ. of Mass-Amherst (2005)
15. Goldstein, S., et al.: The service concept: the missing link in service design research? *Journal of Operations Management* 20(2), 121–134 (2002)
16. Lusch, R.L., Vargo, S.L.: Service-dominant logic: reactions, reflections and refinements. *Marketing Theory* 6, 281–288 (2006)
17. Merkle, B.: Textual modeling tools: overview and comparison of language workbenches. In: *Proceedings of OOPSLA*, pp. 139–148 (2010)
18. McNulty, T., Ferlie, E.: Process transformation: Limitations to radical organizational change within public service organizations. *Organization Studies* 25, 1389–1412 (2004)
19. Peters, B.G., Savoie, D.J.: Managing Incoherence: The Coordination and Empowerment Conundrum. *Public Admin. Review* 56(3), 282–288 (1995)
20. Segelström, F.: Visualisations in Service Design (Licentiate dissertation). Linköping University Electronic Press, Linköping (2010)
21. Papazoglou, M.P., Yang, J.: Design Methodology for Web Services and Business Processes. In: Buchmann, A.P., Casati, F., Fiege, L., Hsu, M.-C., Shan, M.-C. (eds.) *TES 2002. LNCS*, vol. 2444, pp. 54–64. Springer, Heidelberg (2002)
22. UNESCO, Decade of Education for Sustainable Devpt. Report 002166 (2012)
23. United Nations. UN World Urbanization Prospects, Report *ESA/P/WP/224*. United Nations, Department of Economic and Social Affairs, Population Division (March 2012)
24. WHO. Our Cities, Our Health, Our Future. World Health Organization, Center for Health Development. Report 052008 (2008)
25. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M.: Value-Based Service Modeling and Design: Toward a Unified View of Services. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) *CAiSE 2009. LNCS*, vol. 5565, pp. 410–424. Springer, Heidelberg (2009)
26. Wikipedia. Service design, https://en.wikipedia.org/wiki/Service_design (accessed April 15, 2013)
27. Woetzel, J.L., et al.: Preparing for China's Urban Billion. McKinsey Global Institute, Insight Report (February 2009)
28. World Bank. Planning, Connecting, and Financing Cities-Now: Priorities for City Leaders. World Bank, Washington, DC (2013)

Representing and Elaborating Quality Requirements: The QRA Approach

Jie Sun¹, Pericles Loucopoulos², and Liping Zhao³

¹ College of Computer Science, Zhejiang University, Hangzhou, China
jiangbin@zju.edu.cn

² Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece
p.loucopoulos@hua.gr

³ School of Computer Science, University of Manchester, Manchester, U.K.
liping.zhao@manchester.ac.uk

Abstract. This paper presents the quality requirements analysis (*QRA*) approach to requirements modeling. The *QRA* approach supports functional and non-functional requirements modeling in three dimensions: First, it extends functional goal modeling with non-functional goals. Second, it injects QR (quality requirement) specification into business process models. Third, it provides a set of rules for elaborating and refining QRs alongside functional decomposition of business processes. This paper describes *QRA*'s conceptual foundations and illustrates them through goals and business processes of a real world stock trading system.

Keywords: quality requirements (QRs), goal modeling, business process modeling, quality requirements analysis.

1 Introduction

Although quality requirements (QRs) such as reliability, performance, security, compatibility are critically important to software systems, they have often been left out at the early requirements analysis stages and only added to the system at a late stage [1, 2]. According to a recent study [3], this problem might be caused by the significant involvement of system architects in early requirements elicitation, whose concerns are primarily functional requirements, rather than NFRs or QRs. Even when QRs are considered, they are often represented informally, using in-house templates and notations [1].

This paper presents the quality requirements analysis (*QRA*) approach to requirements modeling. The *QRA* approach supports functional and non-functional requirements modeling in three dimensions: First, it extends functional goal modeling with non-functional goals. Second, it injects QR (quality requirement) specification into business process models. Third, it provides a set of rules for elaborating and refining QRs alongside functional decomposition of business processes. This paper describes *QRA*'s conceptual foundations and illustrates them through goals and business processes of a real world stock trading system.

2 Conceptual Foundations of QRA

The *QRA* approach builds on the following three conceptual foundations:

- A *meta-model* that represents the concepts of *QRA* and their relationships.
- A *meta-process* defining the way of working with *QRA*.
- A set of *formulae* for elaborating the QRs.

Due to space limitation, this section will only describe the first two foundations. The description of the third foundation is given in a separate paper [4].

2.1 The *QRA* Meta-model

The meta-model of the *QRA* approach, shown in Fig. 1, defines a set of concepts for representing and elaborating QRs and FRs.

Requirements, in the *QRA* approach, are considered through a prism of two major conceptual artifacts namely *business goals* and *business processes*. These represent the basic building blocks for defining FRs and QRs. Since the *QRA* approach is primarily concerned with QRs, most of the details in the meta-model are about QRs but for completeness purposes, the meta-model also includes a part of FRs definition in order to show the interplay between QRs and FRs.

The *QRA* approach adopts a goal-oriented approach, which in recent years has become a dominant paradigm in RE [5]. Goal modeling involves a series of causal relationships that essentially analyze goals from fuzzy and vague [6] to goals of increasing determinism. Through these causal relationships, goals are organized into a goal hierarchy with the leaves referred to as “*operational goals*” and the others as “*intermediate goals*”. There are two types of causal relationship, one that is concerned with the causality of type “*refine*” between intermediate goals, signifying how a goal, the “*satisficed goal*”, is deconstructed into one or more “*satisfier goal(s)*”, and one that is concerned with the causality between intermediate goals and operational goals, signifying how a goal is “*operationalized*”.

Operational goals are business goals, i.e. customer-facing goals, with exact “*requirements*” for systems intended to satisfy these business goals. Requirements are distinguished into “*FRs*” and “*QRs*”. In *QRA*, FRs and QRs are considered synergistically because there is an inevitable intertwining between them [7].

As shown in the meta-model of Fig. 1, FRs are defined as either the tuple of <Transaction, Business Process, Function Point> concerned with business processes at a macro level or <Transaction, Business Process Element, Function Point>, concerned with individual components of business processes. Both ternary relationships are depicted in the meta-model as the objectified element of “*Function Component*”. QRs are defined symmetrically to FRs as either the tuple <Quality Factor, Business Process, Quality Metric> or <Quality Factor, Business Process Element, Quality Metric>, both of which are shown as the objectified element “*Quality Component*”.

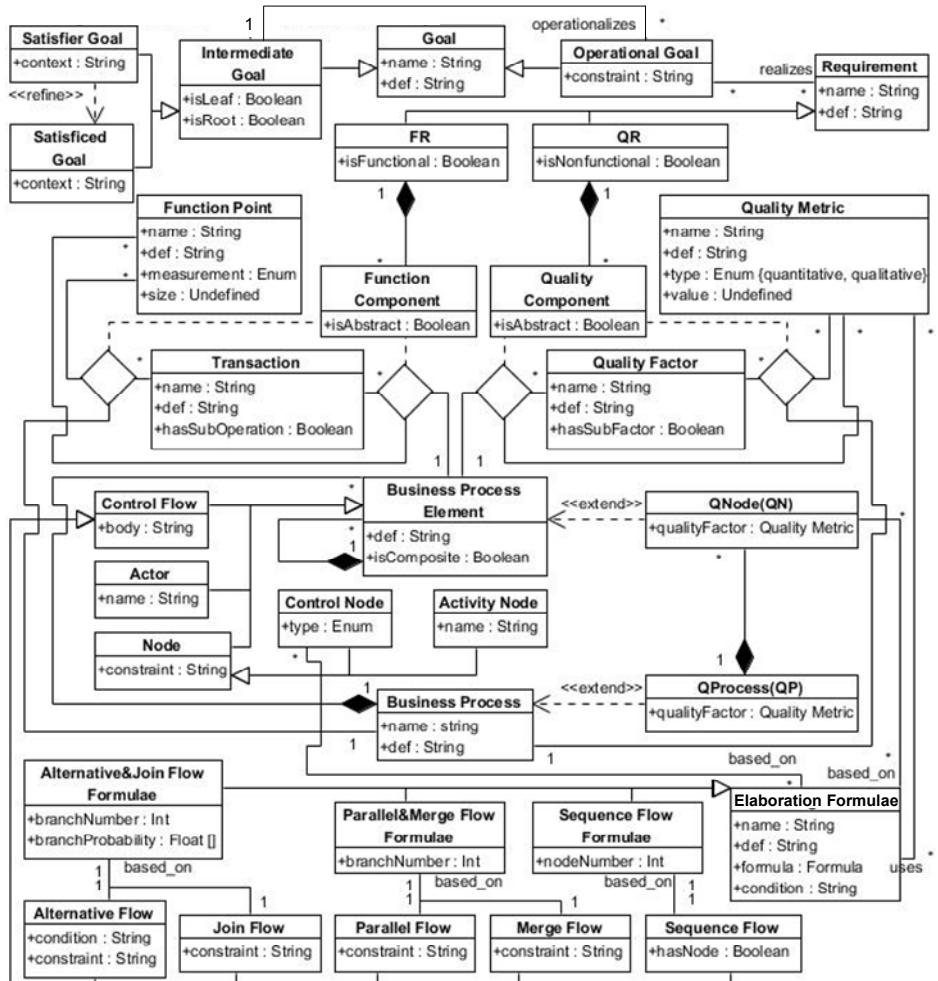


Fig. 1. The QRA meta-model

A business process is a set of ordered behaviors performed by actors to meet specific functional requirements. Generally, there are three types of business process element: i) actors that are responsible for performing behaviors; ii) nodes including activity nodes that specify behaviors and control nodes that coordinate execution flows of activity nodes; iii) control flows that define execution order of nodes and form the basic structure of each business process. There are five types of control flow, namely Sequence Flow, Parallel Flow, Merge Flow, Alternative Flow and Join Flow [8].

Since FRs and QRs are interrelated through business processes, we define an integrated representation of FRs and QRs through two new concepts, QProcess (QP) and Qnode (QN), which extend Business Process and Business Process Element. A QP is a quality-enhanced business process composed of QNs. Each QN

integrates a functional business process element with annotations of quality factors and metrics. Although any business process element can be annotated with QRs, due to space limitation, this paper focuses on the incorporation of QRs into activity nodes.

For each QP as a whole, a set of Elaboration Formulae are defined to measure its QRs, by aggregating the quality metrics annotated on its QNs. According to the five basic control flows of QPs, three types of formulae are defined, which are sequence flow formulae, alternative&join flow formulae and parallel&merge flow formulae. Quality metrics of the overall QP can be calculated by iteratively decomposing the QP into sub-QPs and delegating the calculation to them until each sub-QP can be matched to a basic control flow and calculated by correspondingly formulae.

2.2 The QRA Meta-process

The meta-process of the QRA approach is presented in Fig. 2 and illustrated in Section 3. The QRA meta-process defines the main steps of the approach that involve *goal modeling*, *business process modeling* and *quality requirements elaboration*, making use of the concepts defined in the meta-model of Fig. 1.

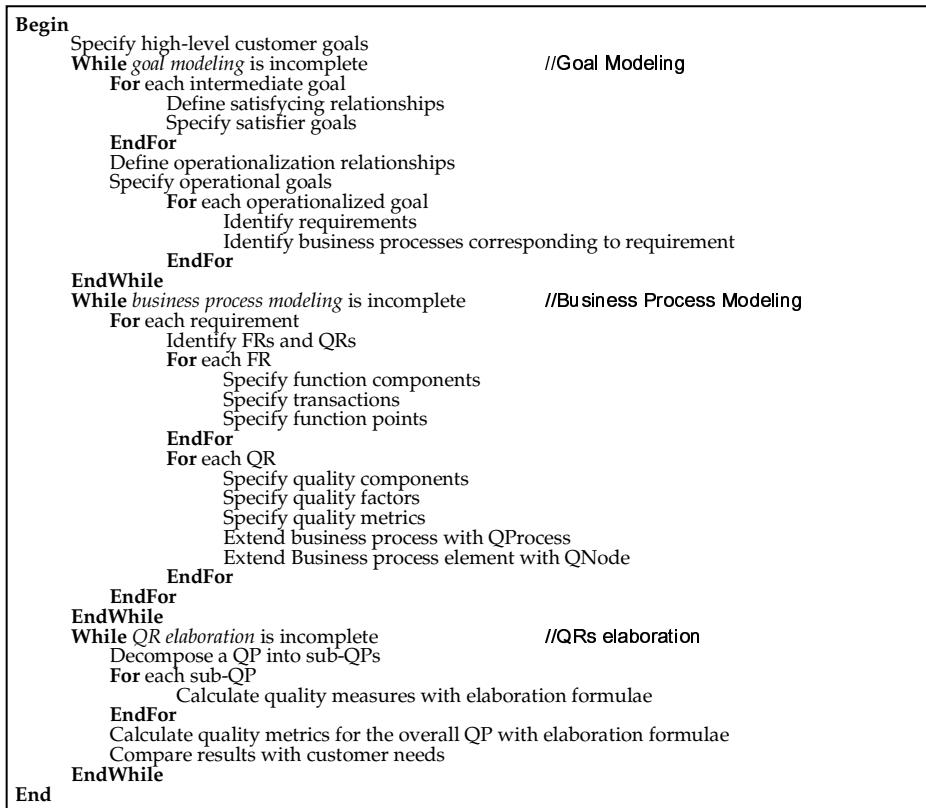


Fig. 2. The QRA meta-process

3 Illustrating the QRA Approach

The concepts introduced in section 2 are illustrated in this section through an example of a stock trading service system (STSS) from a US-based multinational financial firm. STSS provides real-time services to facilitate its users (i.e., traders) to sell or buy shares online and this is realised by interacting with two other agents – Market Data Feeder (MDF) and Printing Agent (PA). STSS needs to fulfill five essential FRs as well as a set of QRs. The five FRs are: (i) Order Entry, (ii) Order Pricing, (iii) Order Matching, (iv) Trade Printing and (v) Trade Allocation. Due to space limitation, this paper only illustrates three key QRs of the system, which are: (i) performance, (ii) recoverability and (iii) accessibility. The following subsections illustrate the three major phases outlined in the *QRA* meta-process of Fig. 2.

3.1 Goal Modeling

According to the *QRA* meta-process, a goal model was constructed to analyze high-level goals to operational ones and from those to define FRs and QRs. A small part of the STSS goal model is shown as Fig. 3. The notation used is an extension of two popular goal modeling techniques namely KAOS [9] and i* [10] and implemented in a prototype tool [11]. The high-level business goal - *to make the firm competitive in the market* - is decomposed into intermediate goals which through successive analysis are defined in terms of the operational goals 1.1.1.1 and 1.1.1.2. In order to realize the two operational goals, we define realization relationships and specify nine requirements, five FRs and four QRs. For QRs, we further identify three relevant quality factors – *recoverability* (QR1), *accessibility* (QR2) and *performance* (QR3&QR4).

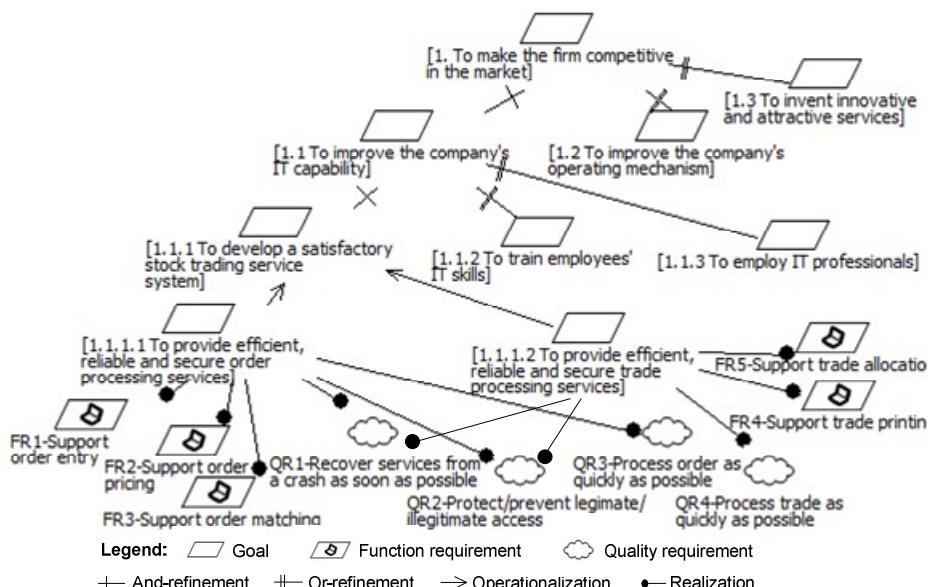


Fig. 3. The Goal Modeling Graph for STSS

3.2 Business Process Modeling

For this example, the business process with its corresponding transactions is shown in Fig. 4 using UML Activity Diagram notation. However, this notation is augmented in order to incorporate the extensions to QProcess and QNode.

In Fig. 4, each transaction is labeled (e.g., T1). Performance, recoverability and accessibility are represented by letter ‘P’, ‘R’ and ‘A’ respectively. If a quality factor (e.g. accessibility) has more than one metric, we need to represent the name of each metric to avoid confusion; otherwise (e.g., performance and recoverability), we only show its quality value. For example, in the *order reception* QN, ‘Is’ means the response time of this transaction is one second; ‘10min’ means the recovery time of this transaction is ten minutes; ‘data protection (90%)’ and ‘access control (90%)’ means the satisfaction percentages of the two accessibility metrics are both 90%. A “Null” value of a quality metric in a certain QN means the metric is irrelevant to this QN. Due to the irrelevance, the satisfaction percentage for a Null value is therefore 100%.

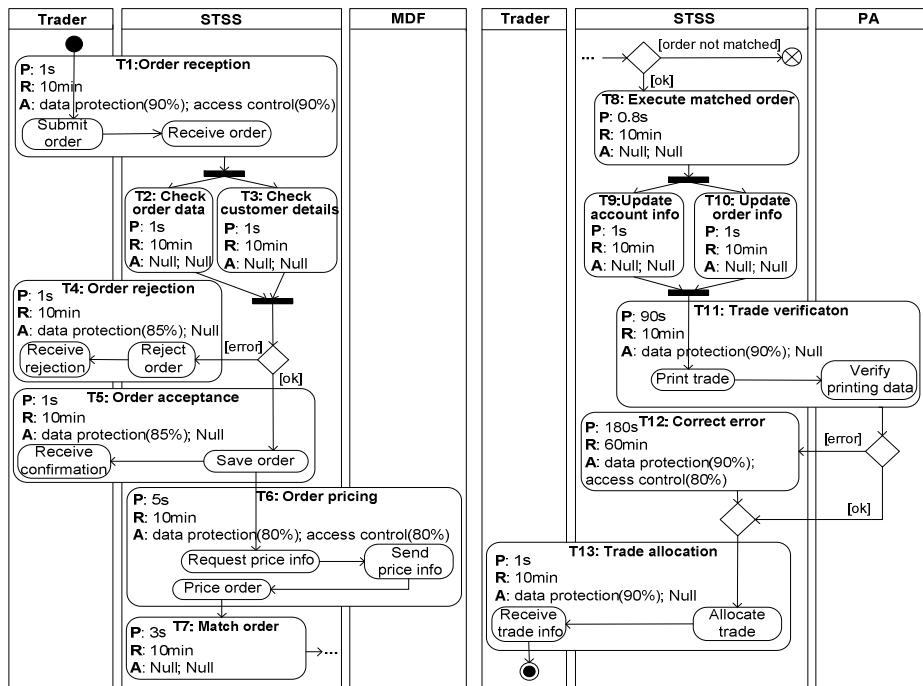


Fig. 4. The QProcess for STSS

3.3 QRs Elaboration

According to the QRA meta-process, the QRs elaboration starts from the iterative decomposition of a QP into sub-QPs until each of which matches one of the five basic process flow patterns. For each matched sub-QP, we apply corresponding formulae to

calculate the quality metrics of its P, R and A. The interested readers are referred to [4] for a detailed description of the QRs elaboration for STSS.

4 Discussion and Conclusion

The *QRA* approach is a significant contribution to the requirements engineering field. In this field, there have been a number of approaches available to support QRs. The NFR Framework [12] is the most well known approach. It treats QRs as softgoals and assists their elicitation, specification and evaluation based on a so-called Softgoal Interdependency Graph (SIG). SIGs treat QRs separately from FRs, whereas in the *QRA* approach, QRs are treated alongside FRs during goal modeling and business process modeling. In order to integrate the NFR Framework with FRs, Chung and Supakkul [13] attempted to combine SIG with functional use cases. The *QRA* approach incorporates QRs into functional business processes.

With respect to the incorporation of QRs into business processes, attempts have been made to represent QRs in different process languages [14, 15]. Specifically, Heidari et al. [16] developed a language-independent business process metamodel enriched with different quality factors and metrics. While these approaches focus on the representation of QRs in business processes, they provide little support for further elaboration of QRs. Aburub et al [17] presented an approach to the identification and inclusion of QRs in business processes. Saeedi et al [18] put forward an approach to measure performance, reliability and cost in BPMN..

Compared with the above approaches, the *QRA* approach provides a more systematic way to representing, elaborating and measuring QRs alongside FRs. Its contribution is three-fold:

The QRA approach enables a systematic representation of QRs and FRs by providing an integrated modeling language. This language, supported by the RE-Tools [11], combines goal model, functional business process model and QR model into one coherent model. The benefit of such a representation is that business analysts can have a consolidated picture of all user requirements, rather than just fragmented requirements. The representational language of *QRA* is therefore an effective communication tool to support the early requirements elicitation and discussion.

The QRA approach facilitates a systematic elaboration of QRs and FRs through three levels of goal modeling, business process and QR modeling. Goal modeling allows business analysts to work with customers in a step-wise refinement fashion by elaborating fuzzy, business goals gradually into concrete, operationalized goals. These concrete goals can then be “designed” into business processes as specific business activities. Finally, QR modeling completes business processes by associating each business activity with a set of related QRs. Such a process serves as a basis for communication and discussion between business analysts and customers as well as a basis for communication between business analysts and software architects.

The QRA approach provides a set of formulae for calculating the values of QRs so that QRs can be measured or quantified. These formulae are based on the five control flow patterns of business process modeling and should be applicable to any business process model. Space limitation prevents us from discussing these formulae in detail.

References

1. Ameller, D., Ayala, C., Cabot, J., Franch, X.: Non-functional Requirements in Architectural Decision Making. *IEEE Software* (2013)
2. Loucopoulos, P., Sun, J., Zhao, L., Heidari, F.: A Systematic Classification and Analysis of NFRs. In: Americas Conference on Information Systems (accepted, 2013)
3. Heesch, U.V., Avgeriou, P.: Mature Architecting—a Survey about the Reasoning Process of Professional Architects. In: IEEE/IFIP Conf. Software Architecture (WICSA 2011), pp. 260–269. IEEE Press, Piscataway (2011)
4. Sun, J., Zhao, L., Loucopoulos, P., Zhou, B.: QRA: A Quality Requirements Analysis Approach for Service Systems. In: International Conference on Services Computing. IEEE Press (2013)
5. Donzelli, P.: A goal driven and agent-based requirements engineering framework. *Requirements Engineering Journal* 9, 16–39 (2004)
6. Bubenko, J., Rolland, C., Loucopoulos, P., DeAntonellis, V.: Facilitating “fuzzy to formal” requirements modelling. In: IEEE International Conference on Requirements Engineering, pp. 154–157. IEEE Press, Los Alamitos (1994)
7. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., Robinson, W.: The Brave New World of Design Requirements. *Information Systems* 36, 992–1008 (2011)
8. Aalst, W.M.P., Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. *Distrib. Parallel Dat.* 14, 5–51 (2003)
9. Lamsweerde, A.: Goal-oriented Requirements Engineering: A Guided Tour. In: IEEE International Symposium on Requirements Engineering, pp. 249–262. IEEE Press (2001)
10. Yu, E.S.K.: Towards modelling and reasoning support for early-phase requirements engineering. In: IEEE International Symposium on Requirements Engineering, pp. 226–235. IEEE Press, Los Alamitos (1997)
11. RE-Tools, <http://www.utdallas.edu/~supakkul/tools/RE-Tools/>
12. Mylopoulos, J., Chung, L., Nixon, B.: Representing and Using Nonfunctional Requirements: A Process-oriented Approach. *IEEE T. Software Eng.* 18, 483–497 (1992)
13. Chung, L., Supakkul, S.: Representing NFRs and FRs: A Goal-oriented and Use Case Driven Approach. In: Dosch, W., Lee, R.Y., Wu, C. (eds.) SERA 2004. LNCS, vol. 3647, pp. 29–41. Springer, Heidelberg (2006)
14. Korherr, B., List, B.: Extending the EPC with Performance Measures. In: ACM Symposium on Applied Computing, pp. 1265–1266. ACM Press, New York (2007)
15. Pavlovski, C.J., Zou, J.: Non-functional Requirements in Business Process Modeling. In: Asia-Pacific Conference on Conceptual Modelling, pp. 103–112. Australian Computer Society, Darlinghurst (2008)
16. 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)
17. Aburub, F., Odeh, M., Beeson, I.: Modelling non-functional requirements of business processes. *Inform. Software Tech.* 49, 1162–1171 (2007)
18. Saeedi, K., Zhao, L., Sampaio, P.R.F.: Extending Bpmn for Supporting Customer-Facing Service Quality Requirements. In: IEEE International Conference on Web Services, pp. 616–623. IEEE Press, Piscataway (2010)

Towards a Strategy-Oriented Value Modeling Language: Identifying Strategic Elements of the VDML Meta-model

Ben Roelens and Geert Poels

Ghent University, Faculty of Economics and Business Administration,
Tweakerkenstraat 2, B-9000 Gent, Belgium
`{Ben.Roelens,Geert.Poels}@UGent.be`

Abstract. The concept of value is increasingly important in organizations. This has led to the creation of value models that capture internal value creation and the external exchange of value between the company and its value network. To facilitate strategic alignment, the meta-model specification of value modeling languages should both fully reflect the strategic choices of a company and define ‘what’ a company must do to realize value creation. In this paper, the Value Delivery Modeling Language (VDML) meta-model elements are assessed by applying these two requirements. The resulting strategy-oriented VDML meta-model perspective is obtained by applying the Design Science methodology, which also includes the use of a case example to demonstrate its utility.

Keywords: value model, strategy orientation, business model, VDML.

1 Introduction

Value modeling gets increasing attention in the field of Conceptual Modeling. These modeling techniques can be used to represent the internal creation of value and the exchange of value between the organization and its external value network. This focus on value creation and exchange is particularly relevant for the alignment of the strategic and the operational layer of the business architecture [1, 2]. Value models can bridge the conceptual gap that exists between strategy formulation, as specified by goal modeling, and the operationalization of a strategy, which is addressed by process models [1, 3, 4]. To this end, value models should represent strategy implementation choices.

As of yet, it has not been investigated whether the current value modeling languages capture strategy implementation, which results in the development of ad-hoc models with a limited strategic modeling scope (confer section 2.2). Consequently, the current value models are not able to effectively support strategic alignment.

To solve this gap, this paper proposes two requirements for value models: *(1) fully reflecting the strategic choices made by the company (i.e., the completeness requirement)* and *(2) using concepts that describe ‘what’ a company must do to create value for itself and its environment, without committing to the required operational details (i.e., the strategy implementation depth requirement)*. These requirements are derived from the business model literature in Management and Organization Science, which is concerned with the strategic logic of an organization, i.e., what a company must do to create value

for itself and its stakeholders. As both value modeling and business modeling are concerned with the study of value creation, the ideas drawn from the knowledge bases of these two research fields can be combined to develop the strategy-oriented value meta-model. The *completeness requirement* is based on the identification of the strategic elements that are needed to describe the implementation of a strategy according to the business model literature. The definition of the business model elements determines the abstraction level (e.g., a black box view on external business partners, individual business processes, and internal organizational structure) that should be adopted by value models to address the *strategy implementation depth requirement*.

The paper is organized as follows. Section 2 reviews related research and the modeling scope of the relevant value modeling languages. Section 3 discusses Design Science as the appropriate methodology for this research. The design and development of the meta-model can be found in section 4, together with the demonstration by means of a case example. Section 5 points at some areas for future research.

2 Background

2.1 Related Research

Previous research, which investigated the relationship between value models and process models, consists of aligning e3-value models with operational models as provided by UML activity diagrams [1, 5-7]. Other research [1, 3, 4, 8] has examined the link between e3-value and strategy models. As these techniques make use of e³-value as value modeling formalism, their focus is on external value exchanges. Indeed, the internal view of e³-value is limited to the value that is exchanged between activities, which offers only a partial view on the internal value creation of companies.

Other research [9] focuses on an enhanced variant of e3-value and process models. Although this research effort takes into account both external value exchange and internal value creation (which distinguishes it from [1, 3-8]), a delineated strategic context, in which value models are applied, is still not present.

2.2 Modeling Scope of Existing Value Modeling Languages

In previous research [10], we reviewed relevant research (i.e., the work of Osterwalder [11], etc.) in Management and Organization Science to identify the elements that provide an integrative view on the business model concept (table 1). This reviewed literature provides a theoretical basis to argue that these elements constitute the set of constructs that should be covered by the intended meta-model (*i.e.*, the *completeness requirement*).

Current value modeling languages only cover loose elements of this framework. The Resource-Event-Agent (*REA*) *Value Chain Specification* [12, 13] represents *resources* and the *value chain* as an enterprise script, which is related to the overall business process architecture. The *REA Value System Level* [13] models the resources that are exchanged between a company and its external environment and corresponds with the *value network* element. The *Value System Level* can also model the *financial structure*, which reflects the monetary flows between the company and its environment. VNA [14] captures the conversion of tangible and intangible assets into value in the context of internal

(e.g., within the company) and external networks (e.g., between the company and its partners). Hence, the meta-model of VNA can capture the *value chain* and the *resources* that are the input to its processes, as well as the *value network* element. Although value models (except VDML; confer infra) do not include competences (e.g., only a pragmatic approach is presented in [15]), *Capability Maps* are well known in management practise. A Capability Map is a representation of ‘what’ a company does to reach its objectives [16], which can be used to model *competences*. E^3 -value [17] offers a representation of the *value proposition* within the context of e-business. To evaluate this value proposition, profitability sheets are used, which include a mathematical calculation of the monetary streams related to the inflow and outflow of value objects. Although this evaluation is linked to the financial structure within the business model, it does not make use of any modeling constructs. E^3 -forces [1] (confer section 2.1) was introduced as a variant of e^3 -value, which explicitly models the strategic perspective of a *value network*.

As *VDM* [2] is the only value model, of which the meta-model is able to cover *all the business model elements*, it is the suitable starting point for this research. However, its meta-model has to be refined as it also contains constructs related to the operational details of customer value delivery and even constructs beyond the scope of the business model elements. Therefore it still needs to be investigated which constructs apply to both the *strategy implementation depth requirement* and the *completeness requirement*.

Table 1. Definition of the constituting elements of the integrative business model framework [10]

Concept	Definition
Resources	Human skills, tangible means, and intangible means under control of an organization by being bought or licensed, which are combined within the value chain of activities.
Value chain	Overall business process architecture that describes the structured set of activities that combine resources to create the necessary competences.
Competence	Ability to coordinate flows of resources through the value chain to realize the intended value proposition.
Distribution channel	The way in which the offering is made available to the customers.
Value proposition	Offered set of products and/or services that provide value to the customers and other partners, and compete in the overall value network.
Value network	Web of relations created with external stakeholders, including suppliers, customers, competitors and partners.
Financial structure	Representation of the costs resulting from acquiring resources, and the revenues in return for the offered value proposition.

3 Methodology

As our research objective is the creation of a strategy-oriented value meta-model, Design Science is the appropriate research methodology. This paper reports on a first iteration of the Design Science Research Methodology process [18]: (1) problem identification and motivation and (2) definition of solution objectives are part of the Introduction section, (3) design and development and (4) demonstration are presented in the next section. A rigorous (5) evaluation is object of future research (confer Discussion section).

The design and development of the intended value meta-model is based on the identification of the VDML meta-model constructs, which apply to both requirements for strategic alignment. To implement the *completeness requirement*, a combination of value modeling languages is used, which separately do not apply to the *completeness requirement*, but collectively cover the seven business model elements. The *strategy implementation depth requirement* will be operationalized by assessing

whether the meta-model constructs (of the combination that collectively applies to the *completeness requirement*) are defined at the right level of abstraction as prescribed by the business modeling literature. Afterwards a mapping between the relevant constructs of the combination of value modeling techniques and the according VDML elements will be performed, based on a comparison of the corresponding definitions.

Demonstration of the developed meta-model is based on the Healthcare Case example [19], which describes the implementation of remote monitoring of high-risk pregnancies within the healthcare industry. As this instantiation is used to demonstrate VDML and its existing notations, it can be used as a benchmark to enable a comparison between the developed meta-model and the VDML models of this case example.

4 Results

4.1 Design and Development

Identification of Relevant Value Modeling Constructs

Resources and Value Chain. The *REA Value Chain Specification* shows the *economic resources* that are input and output of *processes*. As the REA Value Chain adopts a black box view on processes, the meta-model is specified at the right level of abstraction for modeling strategy implementation choices. Also VNA is oriented towards the *deliverables* that are conveyed between organizational roles through *transactions*. As internal roles relate to the internal organizational structure, they are not further included.

Competence. *Capability Maps* represent organizational competences as hierarchies of capabilities that enable value delivery to the customer. This high-level analysis of *competences* provides the right level of abstraction for specifying the value layer within the organization. The detailed decomposition of competences into tasks addresses operational details and is no part of the subsequent mappings.

Value Proposition, Distribution Channel, and Value Network. *Value System Level REA Modeling* is concerned with the *economic resources* that are exchanged between the *enterprise* and its *external business partners*. These elements apply to the *strategy implementation depth requirement* and are further included in the analysis. The meta-model of VNA augments the vision of the REA ontology as it includes the *transactions* through which the *deliverables* are conveyed. Now, the *role* concept within VNA is relevant as it may refer to the company and its external business partners. The meta-model of e^3 -*forces* uses the concepts of *constellation* and *business force* (i.e., *market*) to capture the strategic perspective of a value network. The other elements (i.e., *actor*, *value interface*, *value offering*, *value port*, *value exchange*, and *value object*) originate from e^3 -*value* and model the value exchange between a company and its value network.

Financial Structure. The financial structure can be considered as a specific model view in the *Value System Level REA Modeling* as the relevant revenues and costs can be captured by *monetary resources* that are exchanged by the *enterprise*.

Mapping to VDML Meta-Model Constructs. The mappings between the extracted elements and the VDML meta-model constructs (tables 2-4) extend the mappings

provided in [2]. Corresponding elements of the definitions are characterized by the same layout inside the tables. Mappings that are less straightforward are discussed in the text.

A CapabilityMethod (table 2) is defined as a process at the value layer of the business architecture, which focuses on delivering capabilities (and the resulting value contribution) [2]. However, the concept also includes who is responsible for individual activities, which is outside the scope of a strategy-oriented value modeling language. The definition of a BusinessItem (table 2), which includes anything that is conveyed between two roles, is too narrow to capture resources that are exchanged between black

Table 2. Mapping between the relevant meta-model constructs that address resources, value chain, and competence and the corresponding VDML elements

Value modeling technique	Meta-model element	Definition
REA Value Chain Specification	Economic Resource	Objects that are scarce and have <u>utility</u> and are under the control of an enterprise [12].
VNA	Deliverable	The actual (physical or non-physical) <i>things</i> that move from one role to another [14].
VDML	BusinessItem	Anything that can be acquired or created , that conveys information, obligation or other <u>forms of value</u> and that can be conveyed from a provider to a recipient [2].
VNA	Transaction	Occurrence in which a <i>deliverable</i> , originated by one role, is conveyed to and received by another role [14].
VDML	DeliverableFlow	The transfer of a <i>deliverable</i> from a provider to a recipient [2].
REA Value Chain Specification	Process	The exchange or conversion of an <i>input resource</i> (or set of resources) to an <i>output resource</i> of more value [12].
VDML	CapabilityMethod	A Collaboration specification that defines the Activities, DeliverableFlows, BusinessItems, CapabilityRequirements and Roles that deliver a Capability and associated value contributions [2].
Capability Maps	Competence	Network of valuable capabilities (i.e., the <i>ability to make use of resources to perform some task or activity</i>) in terms of enabling the firm to deliver a fundamental customer benefit [15].
VDML	CapabilityOffer	The <i>ability to perform a particular kind of work</i> and deliver desired value , by applying resources that are managed together, possibly based on formalized methods [2].

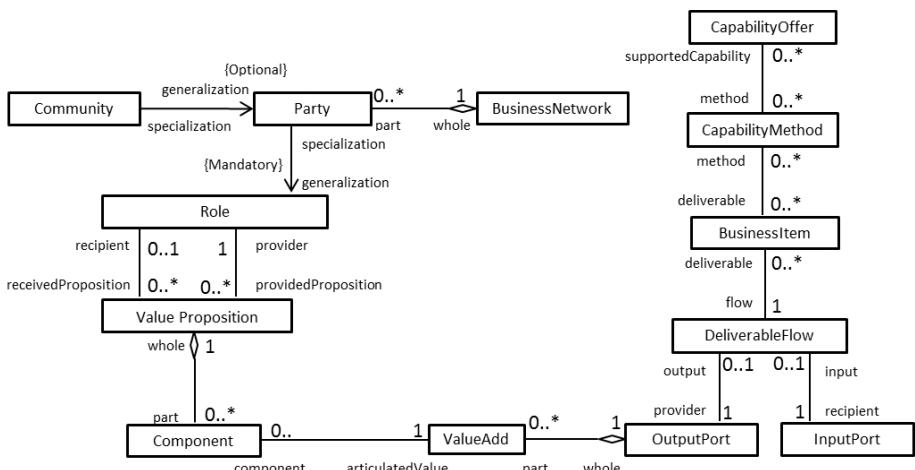


Fig. 1. Strategy-oriented VDML meta-model

box processes (i.e., which abstract from internal roles). Within the meta-model (figure 1), these problems can be overcome as internal roles are omitted and BusinessItems are only specified as deliverables for CapabilityMethods. Moreover, the internal activities, which constitute these CapabilityMethods, are not explicitly included.

The financial structure (table 4) results from the whole of monetary streams to and from a company. Within VDML, an Economic Resource is modeled as a BusinessItem, while the flow of these resources is represented by a DeliverableFlow. However, the combination of these elements is too general to capture the financial structure, so the difference between revenues and costs should be stored as a profit attribute of a Party.

Table 3. Mapping between the relevant meta-model constructs that address value proposition, distribution channel, and value network and the corresponding VDML elements

Value modeling language	Meta-model element	Definition
Value System Level REA Modeling	Enterprise, External Business Partner	Actors in the value system such as the company , its suppliers , its customers , its creditors/investors , and its employees [13].
VNA	Role	Real people or participants (both individuals and organizations) in the network who provide contributions and carry out functions [14].
e ³ -forces	Constellation	Coherent set of two or more actors who cooperate to create value to their environment [1].
VDML	Party	Roles specific to and contained in the BusinessNetwork [2].
e ³ -forces	Market	Set of organizations operating in the environment of a constellation [1].
VDML	Community	A loose collaboration of participants with similar characteristics or interests [2].
e ³ -forces	Value Interface	Group of one ingoing and one outgoing value offering, which shows the mechanism of economic reciprocity [17].
VDML	No specific concept	The aggregate of value propositions provided and received by one party in a business network [2].
e ³ -forces	Value Offering	Models what an actor offers to or requests from its environment , which is a set of equally directed value ports [17].
VDML	Value Proposition	Expression of the values offered to a recipient evaluated in terms of the recipient's level of satisfaction [2].
	Component ValueAdd	The components that constitute the ValueProposition [2]. Objects that represent the values that are delivered by a <i>OutputPort</i> [2].
e ³ -forces	Value Port	A means of an actor to show to its environment that it wants to provide or request value objects [17].
VDML	Input Port Output Port	Receives the deliverable that is transferred via the DeliverableFlow [2]. Provides the deliverable that is transferred via the DeliverableFlow [2].
e ³ -forces	Value Exchange	Connection between two value ports, which represents that two actors are willing to exchange value objects with each other [17].
VNA	Transaction	Occurrence in which a deliverable , originated by one role , is conveyed to and received by another role [14].
VDML	DeliverableFlow Channel	The transfer of a deliverable from a provider to a recipient [2]. Mechanism to execute a deliverable flow [2].
Value System Level REA Modeling	Economic Resource	Objects of economic value (with or without physical substance) that are provided or consumed by an enterprise [13].
VNA	Deliverable	The actual (physical or non-physical) things that move from one role to another [14].
e ³ -forces	Value Object	Product and services that are of value for one or more actors [1].
VDML	BusinessItem	Anything that can be acquired or created, that conveys information, obligation or other forms of value and that can be conveyed from a provider to a recipient [2].

Table 4. Mapping between the relevant meta-model constructs that address financial structure and the corresponding VDML elements

Value modeling language	Meta-model element	Definition
Value System Level REA Modeling	Money	<i>Monetary Objects of economic value</i> (with or without physical substance) that are provided or consumed by an enterprise [13].
	Enterprise	Actor in the value system [13].
VDML	Party_profit	The <i>difference between revenue and cost</i> that a providing Party in a BusinessNetwork might realize [2].
	DeliverableFlow	The <i>transfer of a deliverable</i> from a provider to a recipient [2].
	BusinessItem	<i>Anything</i> that can be acquired or created, that conveys information, obligation or other <i>forms of value</i> and that can be conveyed from a provider to a recipient [2].

4.2 Demonstration

This section makes an analysis of the meta-model elements of the nine VDML modeling viewpoints used in the Healthcare Case Example [19]. These elements will be compared with the developed meta-model to demonstrate the utility of our proposal. This way of working is needed as we do not dispose yet of a graphical notation to visualize instantiations of the strategy-oriented meta-model within VDML.

(1) *Value Proposition Exchange Diagrams* visualize *Value Propositions* that are offered between *Party Roles* in the *BusinessNetwork*. (2) *Role Collaboration Diagrams* focus on the *DeliverableFlows* of *BusinessItems* between *Parties*, as well as *sequence numbers* to identify the exchange order of these *BusinessItems*. (3) A *Business Network Activity Diagram* identifies the *high-level Activities* that need to be executed by *Parties* and the *BusinessItems* (held in *Stores*) that flow between these activities. These DeliverableFlows can be split by an *alternative deliverable output*. (4) A *Capability Method Delegations Diagram* identifies the internal *CapabilityMethods* that realize the *high-level Activities* of the different *Party Roles*. (5) *Capability Method Diagrams* define the *Activities* needed to realize a *CapabilityMethod*, the associated flows of *BusinessItems*, and the responsible internal *Roles*. (6) A *Measurement Dependency Graph* augments this view by making explicit the *value contribution* of individual *Activities*. (7) A *Capability Management Diagram* shows the *CapabilityOffers*, as well as the required *CapabilityMethods* and *Resources*, performed by an *Organization Unit*. (8) A *Capability Map* identifies *Capabilities* that can be improved. (9) An *Organization Structure Diagram* models the internal structure as a hierarchy of *Organization Units*.

The analysis of the VDML modeling viewpoints provides an insight of the strengths of the developed meta-model. Although the VDML meta-model applies to the *completeness requirement*, the internal view (perspectives 1-3) and the external view (perspectives 4-9) on strategy implementation are addressed by separate models. This is solved by our proposal, which fully addresses the strategic scope within a single model view. Moreover, existing VDML views adopt operational details in their instantiations (i.e., (5) internal activities, (5, 7) internal responsibilities, (9) internal structure, etc.). The developed meta-model, which is based on the *implementation depth requirement*, makes abstraction from any of these operational details. Hence, instantiations of the developed meta-model have the potential to effectively support strategy orientation.

5 Discussion

Although the utility of the strategy-oriented meta-model is demonstrated in the previous section, future research is required. A next step is developing a graphical notation to visualize the strategy-oriented VDML meta-model. This will enable the creation of model instantiations, which can be rigorously evaluated. Moreover the developed meta-model needs to be further formalized to ensure the development of instantiations, which comply with the meta-model restrictions. This will result in the creation of a proper value modeling language that captures the value layer in an organization.

Afterwards, the alignment between the strategy, value, and operational layer of the business architecture needs to be ensured. This will require a comparison of the existing alignment techniques (confer section 2.1) to discover which of these research efforts is suitable to realize strategic alignment and which eventual adaptations are needed.

References

1. Pijpers, V., de Leenheer, P., Gordijn, J., Akkermans, H.: Using Conceptual Models to Explore Business-ICT Alignment in Networked Value Constellations. Requirements Engineering 17(3), 203–226 (2012)
2. OMG: Value Delivery Modeling Language (VDML), bni/2012-11-06 (2012)
3. Gordijn, J., Petit, M., Wieringa, R.: Understanding Business Strategies of Networked Value Constellations Using Goal- and Value Modeling. In: 14th IEEE International Requirements Engineering Conference, RE 2006, Minneapolis/st Paul, Minnesota, pp. 129–138 (2006)
4. Andersson, B., Johannesson, P., Zdravkovic, J.: Aligning Goals and Services through Goal and Business Modelling. Information Systems and E-Business Management 7(2), 143–169 (2009)
5. Zlatev, Z., Wombacher, A.: Consistency between E3value Models and Activity Diagrams in a Multi-Perspective Development Method. In: Meersman, R., Tari, Z. (eds.) OTM 2005. LNCS, vol. 3760, pp. 520–538. Springer, Heidelberg (2005)
6. Edirisuriya, A., Johannesson, P.: On the Alignment of Business Models and Process Models. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008 Workshops. LNBP, vol. 17, pp. 68–79. Springer, Heidelberg (2009)
7. Andersson, B., Bergholtz, M., Grégoire, B., Johannesson, P., Schmitt, M., Zdravkovic, J.: From Business to Process Models – a Chaining Methodology. In: Pigneur, Y., Woo, C. (eds.) Proceedings of the CAiSE*06 Workshop on Business/IT Alignment and Interoperability (BUSITAL 2006). CEUR-WS, vol. 237, pp. 1–8 (2006)
8. Gordijn, J., Yu, E., van der Raadt, B.: E-Service Design Using I* and E3value Modeling. IEEE Software 23(3), 26–33 (2006)
9. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T.: On the Notion of Value Object. In: Dubois, E., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 321–335. Springer, Heidelberg (2006)
10. Roelens, B., Poels, G.: Towards an Integrative Component Framework for Business Models: Identifying the Common Elements between the Current Business Model Views. In: Deneckère, R., Proper, H. (eds.) CAiSE 2013 Forum at the 25th International Conference on Advanced Information Systems Engineering. CEUR-WS, vol. 998, pp. 114–121 (2013)

11. Osterwalder, A., Pigneur, Y., Tucci, C.: Business Model Generation: A Handbook for Visionaries. In: Game Changers, and Challengers, John Wiley and Sons Inc., Hoboken (2010)
12. Geerts, G., McCarthy, W.: An Ontological Analysis of the Economic Primitives of the Extended-REA Enterprise Information Architecture. International Journal of Accounting Information Systems 3, 1–16 (2002)
13. Dunn, C., Cherrington, J., Hollander, A.: Enterprise Information Systems: A Pattern-Based Approach. McGraw-Hill Irwin, Boston (2005)
14. Allee, V.: Value Network Analysis and Value Conversion of Tangible and Intangible Assets. Journal of Intellectual Capital 9(1), 5–24 (2008)
15. Hafeez, K., Zhang, Y., Malak, N.: Determining Key Capabilities of a Firm Using Analytic Hierarchy Process. International Journal of Production Economics 76(1), 39–51 (2002)
16. Cook, D.: <http://msdn.microsoft.com/en-us/library/bb402954.aspx>
17. Gordijn, J., Akkermans, H.: Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas. Requirements Engineering Journal 8(2), 114–134 (2003)
18. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2007)
19. OMG: VDML Healthcare Use Case, bmi/2012-11-11 (2012)

Ontological Distinctions between Means-End and Contribution Links in the *i** Framework

Renata S.S. Guizzardi¹, Xavier Franch², Giancarlo Guizzardi¹, and Roel Wieringa³

¹ Ontology & Conceptual Modeling Research Group, UFES, Brazil

{rguizzardi, gguizzardi}@inf.ufes.br

² Universitat Politècnica de Catalunya (UPC)

Barcelona, Spain

franch@essi.upc.edu

³ University of Twente

Enschede, The Netherlands

roelw@cs.utwente.nl

Abstract. The *i** framework is a renowned Requirements Engineering approach. This work is part of an ongoing effort to provide ontological interpretations for the *i** core concepts. With this, we aim at proposing a more uniform use of the language, in a way that it can be more easily learned by newcomers and more efficiently transferred to industry. Our approach is based on the application of a foundational ontology named UFO, which is used as a semantically coherent reference model to which the language should be isomorphic. In this paper, we focus on the *Means-end* and the *Contribution* links. We aim at presenting the community with some possible ontological interpretations of these links, aiming at promoting constructive debate and receiving feedback about the validity of our assumptions.

1 Introduction

The *i** framework [1] is a renowned Requirements Engineering approach that has been alive for almost two decades, continuously attracting new interest both in academia and industry. The community that develops *i** is relatively big and these developers, who are geographically dispersed, tend to ascribe different (and sometimes conflicting) meanings to its constructs. The diversity and looseness of the defined variants and extensions could be considered a barrier for one to learn how to use the language as most of the times, it is hard to grasp when a specific construct can and should be used and when it should not (e.g., see empirical study on the *is-a* construct [2]). It is our belief that this hampers the efficient communication of knowledge among experts of the community [3], the learning curve of newcomers, and the adoption of the framework by practitioners.

In the past few years, the community has become aware of this problem and several attempts have been made for facilitating the access and uniform use of the *i** language. Works on metamodeling have tried to make it clear the meaning ascribed to

the distinct constructs [4][5][6]. Although we recognize there are significant outcomes of these works (e.g. pointing out the applied concepts in particular variations; showing the author's view on how concepts relate), these attempts did not quite succeed in providing interoperability, simply because metamodels are powerful structures to define a language's syntax while being very limited in terms of clarifying its semantics. Cares [7] has proposed an interoperability method that considers a supermetamodel [8], which facilitates the translation from an i^* variant to another, and an XML-based mark-up language, named iStarML [9], which triggers existing tools to interoperate as much as their underlying metamodel allows. This approach has advanced the state of the art, by providing a standard interoperability format that facilitates model translation, but we are afraid that iStarML only makes syntactic checks, leaving the semantic interoperability issues still untouched.

Our approach goes beyond the work of Cares, aiming at defining a common ontology for the core concepts of the language. In this paper, our focus is to propose distinctions between the *Means-end* and the *Contribution* links, often present in different languages with closely related but ambiguous meanings. It is our goal to present the community with some possible ontological interpretations of these distinctions, aiming at promoting constructive debate and receiving feedback about the validity of our assumptions.

Ontologies have been used to establish a common understanding regarding a domain of interest, functioning as an interchange language for communication between applications and organizations. Guizzardi [10] proposes a method to evaluate and (re)engineer modelling languages. Such method has been applied in the context of the i^* framework [10][12]. It consists on using an ontology as a reference model and trying to make the metamodel of the language isomorphic to this ontology. In this way, the language is said to represent well the domain described by the ontology.

In this work, we apply the UFO ontology [10,11] as the reference model. Our choice for UFO was motivated by our in depth knowledge of its foundations, but also by the substantial record of its successful use to analyze, redesign and integrate languages and reference models in a large number of domains. These analysis initiatives have targeted significant approaches in the literature such as BPMN, ARIS, REA, ITIL, RM-ODP, AORML, UML, Archimate, among others. In particular, in a number of publications, we have used UFO to make explicit the ontological semantics underlying i^* and TROPoS [11,12]. Finally, in the overall research project of analyzing i^* , it is fundamental be able to count on a foundational ontology that elaborates on ontological distinctions among object types and that embeds a fuller ontological theory of relations. This requirement makes UFO a more suitable choice for this purpose than alternatives such as GFO and DOLCE.

The remainder of this paper is organized as follows: Section 2 explains the concepts of the UFO foundational ontology which form the basis for the new ontological definitions described in this paper; Section 3 proposes the formal ontological definitions of the Means-end and Contribution links; at last, section 4 discusses whether and if so, what we have contributed to the goal of making i^* more usable and useful.

2 The Supporting UFO Concepts

Before analyzing the *i** Means-end (ME) and Contribution links, it is important to provide an ontological view of the language intentional elements. We do that by analyzing the language elements in terms of the ontological distinctions put forth by the foundational ontology UFO [11][12].

In UFO, a stakeholder is represented by the Agent concept, defined as a concrete Endurant (i.e. an entity that endures in time while maintaining its identity) which can bear certain Intentional States. These intentional states include Beliefs, Desires and Intentions. Intentions are mental states of Agents which refer to (are about) certain Situations in reality. Situations are snapshots of reality which can be understood as a whole and which can be actual or counterfactual. If a Situation is actual in a certain Time Interval, we say that the situation obtains-in that time interval. Situations are composed of Endurants. In particular, a situation can be composed of sub-situation. If a Situation obtains-in a certain Time Interval then all its constituents exist in that Time Interval.

Intentions have propositional-contents that can be true according to the way the world happens to be (i.e., which Situations actually obtain-in the world in a certain Time Interval). For instance, suppose that I believe that London is the capital of England. This Belief is correct if its propositional-content is true which, in turn, is the case iff there is a city called London and a country called England and the former bears a legal relation of being the capital of to the latter. The propositional-content (i.e., proposition) of an Intention is termed a Goal.

We here take propositions to be abstract ontological entities that bear a relation of *satisfiability* to situations in the world. Thus, when writing *satisfies(s, p)* we mean that *p* is true iff the situation *s* obtains-in reality. We also define the predicate *implies(p, p')* holding between propositions meaning: (1) $\text{implies}(p, p') \leftrightarrow (\forall s \text{ Situation}(s) \wedge \text{satisfies}(s, p) \rightarrow \text{satisfies}(s, p'))$. The set of situations which satisfy a Goal *G* is termed *G's satisfiability set*.

In contrast to Endurants, Events are perduring entities, i.e., entities that occur in time, accumulating their temporal parts. Events are triggered-by certain Situations in reality (termed their pre-situations) and they change the world by producing a different post-situation. Actions are Events deliberately performed by Agents in order to fulfil their Intentions. So, we state that if an Intention *i* of Agent *X* is the reason for the performance of Action *a* (by *X*) then *X believes* that *a* has as post-situation *S* which satisfies the propositional-content of *i*. In case *X's belief* is correct then we say that Action *a* achieves Goal *G*, i.e.: (2) $\text{achieves}(a, G) \leftrightarrow \text{satisfies}(\text{post-situation}(a), G)$. In other words, an Action achieves a Goal if the Action brings about a Situation in the world which satisfies that Goal.

We here interpret *i** goals, tasks and agents as their counterparts in UFO (with Action as task). As such, we interpret the so-called goal *and-decomposition* and goal *or-decomposition* in *i** as follows: (3) a goal *G* is *and-decomposed* in goals *G₁...G_n* iff *G* is satisfied by exactly those situations which satisfy *G₁...G_n* conjunctively, i.e., the situations which satisfied all *G₁...G_n*; (4) a goal *G* is *or-decomposed* in goals

$G_1 \dots G_n$ iff G is satisfied by exactly the situations which satisfy the goals $G_1 \dots G_n$ disjunctively, i.e., by any situation which satisfy each of the goals $G_1 \dots G_n$. In the formulae above, we assume that goals $G_1 \dots G_n$ are distinct and all of these goals are informative (i.e., they are possibly achieved).

3 Means-End Links vs. Contribution Links

This section is dedicated to provide ontological interpretations for Means-end links and Contribution links in i^* , especially by comparing and contrasting each other. To distinguish the four Contribution values, we chose to use the GRL syntax of Make and Break (also present in the i^* wiki), because these names express more clearly the semantic distinctions among the values, when compared to the traditional i^* values of ++ and --. The analysis of the partial contribution links (Help and Hurt) remains as future work. Figure 1 presents two cases which will assist us in the definitions of the links, presented in the following two subsections.

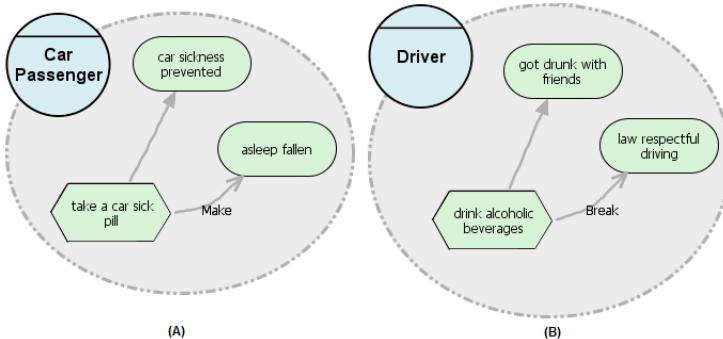


Fig. 1. Examples in i^* used to differentiate the Means-end and Contribution links

3.1 Means-End Link vs. Make Contribution Link

Figure 1A) shows an example, in which a Car Passenger¹ agent executes the take a car sick pill task in order to prevent himself from being sick during the journey he is making (Means-end link to car sickness prevented goal). As a side effect, the Car Passenger also goes to sleep (Make Contribution link to asleep fallen goal).

Both goals depicted in the model are equally accomplished: following the proposal in [12], we here assume that the Means-end link leads to full accomplishment and the Contribution link value is Make, which also indicates the goal is completely fulfilled. So then, what is the distinction among these two links? We here claim that the difference is given by the Intention behind the execution of the task.

¹ From now on, we use a different font for the names of the instances of the i^* actors and intentional concepts, such as goals, tasks, and resources.

As shown in Section 2, as result of the mapping from *i** tasks into UFO actions, every task is associated with a causing intention whose propositional content is a goal. In other words, we execute a particular task in order to accomplish a specific goal. In *i**, the association between the task and the goal in this case is made by a Means-end link (e.g. *take a car sick pill task* as means to *car sickness prevented* goal). On the other hand, this same task can also generate some other goals to be accomplished, without however, being intended by the choice of this particular task. In this case, a Make Contribution link is established (e.g. *take a car sick pill task* as means to *asleep fallen* goal). From here on, we focus on a few logical statements that help defining these notions. Firstly, we define the notion of deliberately achieving a goal as follows: (5) $\text{deliberately-achieves}(a, G) \leftrightarrow \text{achieves}(a, G) \wedge (\exists i \text{ intention}(i) \wedge \text{is-reason-for}(i, a) \wedge \text{implies}(\text{propositional-content}(i), G))$.

In other words, an action *a* deliberately achieves a goal *G* iff this action achieves *G* (i.e., causes the world to be in a state which makes *G* true) but also this action must be motivated by an intention (intention *i* is the reason for action *a*) whose propositional content implies *G* (in the sense defined in Section 2). By using this notion, we clarify the distinction between Means-end links (ME) and Make Contribution links: (6) $\text{action}(a) \wedge \text{goal}(G) \wedge \text{ME}(a, G) \rightarrow \text{deliberately-achieves}(a, G)$; (7) $\text{action}(a) \wedge \text{goal}(G) \wedge \text{MakeCont}(a, G) \rightarrow \text{achieves}(a, G) \wedge \neg\text{deliberatively-achieves}(a, G)$.

Now, it is important to analyze what are the changes in these relations for the case of having softgoals as ends. To differentiate between a goal and a softgoal, UFO considers the relation of satisfaction as a ternary relation that can hold between an Agent, a Goal and a *goal satisfiability set*. An instance of this relation is derived from the consideration (*Belief*) of an agent that a particular set of situations satisfies the goal at hand. In fact, it is exactly this concept which seems to capture the aforementioned notion of *softgoals* and its difference w.r.t. *hardgoals*: a goal *G* is said to be a *softgoal* iff it is possible that two rational agents *X* and *Y* differ in their beliefs to which situations satisfy that goal. In other words, we say that a *softgoal* is satisfied according to the *Belief* of a particular *Agent*, while *hardgoals* are either satisfied or not, with no need to consider the agent or her belief [12].

Following the analysis of the given example, what if instead of a hardgoal, Figure 1A) presented a softgoal, e.g. a *feel well* softgoal instead of the *car sickness prevented* hardgoal. Then the analysis would be in place. Firstly, for the case of softgoals, the predicate *satisfies* must be agent-indexed, i.e., it must be defined as a ternary predicate *satisfies(a, G, X)*. We can then redefine the predicates *achieves* and *deliberatively-achieves* for the case of a softgoal. Both redefined predicates now also take into account a third argument, namely, an agent *X*: (8) $\text{achieves}(a, G, X) \leftrightarrow \text{satisfies}(\text{post-situation}(a), G, X)$; (9) $\text{deliberatively-achieves}(a, G, X) \leftrightarrow \text{achieves}(a, G, X) \wedge (\exists i \text{ intention}(i) \wedge \text{is-reason-for}(i, a) \wedge \text{implies}(\text{prop-cont}(i), G))$. The relation of Means-End links is then redefined accordingly: (10) $\text{action}(a) \wedge \text{softgoal}(G) \wedge \text{agent}(X) \wedge \text{ME}(a, G, X) \rightarrow \text{deliberatively-achieves}(a, G, X)$. Finally, supposing the softgoal was the contribution target (e.g. a *got asleep fast* softgoal instead of the *asleep fallen* hardgoal), a similar definition can now be provided for the case of Make Contributions: (11) $\text{action}(a) \wedge \text{softgoal}(G) \wedge \text{agent}(X) \wedge \text{MakeCont}(a, G, X) \rightarrow \text{achieves}(a, G, X) \wedge \neg\text{deliberatively-achieves}(a, G, X)$

In other words, when we deal with softgoals as ends, both in the case of Means-end and Contribution, the only existing distinction regards the fact that satisfiability sets for goals are indexed to particular agents.

Table 1 summarizes the difference between ME and Make Contribution links.

Table 1. Guidelines: Means-end link vs. Make Contribution link

<ul style="list-style-type: none"> • Action a --- means-end \rightarrow hardgoal G for an actor A iff <ol style="list-style-type: none"> 1. By choosing to perform a, it was A's intention to achieve goal G, 2. Performing a causes situation S and 3. Situation S satisfies G, • For softgoal G, replace 2 and 3 by <ol style="list-style-type: none"> 2. Performing a causes situation S according to A and 3. Situation S satisfies G according to A
<ul style="list-style-type: none"> • Action a --- make contribution \rightarrow hardgoal G for an actor A iff <ol style="list-style-type: none"> 1. By choosing to perform a, it was NOT A's intention to achieve goal G, 2. Performing a causes situation S and 3. Situation S satisfies G, • For softgoal G, replace 2 and 3 by <ol style="list-style-type: none"> 2. Performing a causes situation S according to A and 3. Situation S satisfies G according to A

3.2 Break Contribution

The example of Figure 1B) presents the following situation: a **Driver** agent drinks a high dosage of alcohol (**drink alcoholic beverages** task) in order to **got drunk with friends**. However, this contributes negatively (break-valued contribution link) to his other goal to **law respectful driving**.

In UFO, a situation *triggers* an action if the obtaining of that situation enables the action to occur. In other words, the situation is the *pre-situation* of that Action. We say that a situation S *disables* an action a if that situation conflicts with the situation S' which triggers a (the *pre-situation* of a). Having these definitions at hand, we are able to define break contribution. Now, we have also the notion of conflicting situations which can be characterized as follows: (12) $\text{conflicts}(S, S') \rightarrow (\forall t \text{ time-interval}(t) \rightarrow \neg(\text{obtains-in}(S, t) \wedge \text{obtains-in}(S', t)))$. In other words, two situations conflict if they cannot obtain in the same time. We say that a situation S *disables* an action a if that situation conflicts with the situation S' which triggers a (the *pre-situation* of a): (13) $\text{disable}(S, a) \leftrightarrow (\exists S' \text{ triggers}(S', a) \rightarrow \text{conflicts}(S, S'))$. Finally, we can define the break contribution (*BreakCont*) relation between an action and a goal as follows: (14) $\text{action}(a) \wedge \text{goal}(G) \wedge \text{BreakCont}(a, G) \rightarrow \exists S \text{ Situation } (S) \wedge (\text{post-situation}(a) = S) \wedge (\forall a' \text{ Action}(a') \wedge \text{achieves}(a', G) \rightarrow \text{disables}(S, a'))$.

In other words, we state that an action a breaks a goal G iff that action brings the world to a state such that while that state persists, no action which can bring about G can possibly be performed. In reality, if a situation like the one depicted in Figure 3 is modeled in i^* , then the action represented by the **drink alcoholic beverages** task disables all actions which would lead to the **law respectful driving**, i.e., while the

situation created by that action persists (the person at hand being drunk), one cannot perform any action of driving which satisfies the goal of driving legally.

Again, we remind the reader that if the end of the contribution were a softgoal, the only distinction regarding the BreakCont definition would be adding the agent in the loop, as follows: (15) $\text{action}(a) \wedge \text{goal}(G) \wedge \text{agent}(X) \wedge \text{BreakCont}(a, G, X) \rightarrow \exists S \text{ Situation}(S) \wedge (\text{post-situation}(a) = S) \wedge (\forall a' \text{ Action}(a') \wedge \text{achieves}(a', G, X) \rightarrow \text{disables}(S, a'))$.

Table 2 presents some guidelines for the use of the Break Contribution link.

Table 2. Guidelines for the use of the *i** break contribution link

- | |
|---|
| <ul style="list-style-type: none"> • Action a --- break contribution \rightarrow hardgoal G for an actor A iff <ol style="list-style-type: none"> 1. Performing a causes situation S and 2. S disables any action a' that satisfies G. • For softgoal G, replace 2 by <ol style="list-style-type: none"> 2. S disables any action a' that satisfies G according to A. |
|---|

4 Conclusions

In this paper, we report the latest results of a long term research effort in creating a common ontology for the *i** core elements. With this work, we hope to contribute to the ongoing effort of the *i** community to clarify the specific uses for each of the framework's constructs, as well as to promote interoperability among the distinct *i** variants. In [10] we investigated the semantics of the *i** intentional elements (e.g. hardgoal, softgoal, resource, etc.) while in [12] we focused in understanding the difference between the Means-end link and the decomposition relations. In this particular paper, we presented ontological distinctions between the Means-end and Contribution links, having actions as means and goals and softgoals as ends. For that, we analyzed these links in light of the UFO foundational ontology. For each analysis made, we provided illustrative examples and logical statements to formalize our findings. Relevant results of this initiative are both an understanding of the involved constructs and some knowledge that *i** modelers can use as rationale when constructing their diagrams. However, it is clear that a proposal of this nature cannot be limited to the initiative of particular researchers. A fundamental part of our agenda is to promote the discussion of the ontology foundation of *i** at the community level.

From a scientific perspective, the next step of this work is to switch the means and the ends of the investigated links, considering other types of intentional elements as means and ends. In fact, in the logical statement of this paper, we type the means and the ends (e.g. $\text{action}(A)$, $\text{goal}(G)$, etc.) as an attempt to make these formulas as flexible as possible, having this next step in mind. And in addition to completing the analysis started here, investigating the remaining *i** links is also part of our research agenda.

Acknowledgements. This work has been partially supported by the Spanish project TIN2010-19130-C02-00. We are also grateful to the support provided by FAPES (PRONEX #52272362/2011) and CNPq Productivity Grant #311578/2011-0). Finally, we thank Lidia López for her help in designing the figures presented in this paper.

References

1. Yu, E.: Modeling Strategic Relationships for Process Reengineering. PhD thesis, University of Toronto, Canada (1995)
2. López, L., Franch, X., Marco, J.: Specialization in *i** Strategic Rationale Diagrams. In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012. LNCS, vol. 7532, pp. 267–281. Springer, Heidelberg (2012)
3. López, L., Franch, X., Marco, J.: Making Explicit some Implicit *i** Language Decisions. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 62–77. Springer, Heidelberg (2011)
4. Amyot, D., Horkoff, J., Gross, D., Mussbacher, G.: A Lightweight GRL Profile for *i** Modeling. In: Heuser, C.A., Pernul, G. (eds.) ER 2009. LNCS, vol. 5833, pp. 254–264. Springer, Heidelberg (2009)
5. Susi, A., Perinni, A., Mylopoulos, J., Giorgini, P.: The Tropos Metamodel and its Use. *Informatica* 29, 401–408 (2007)
6. Lucena, M., Santos, E., Silva, C., Alencar, F., Silva, M.J., Castro, J.: Towards a Unified Metamodel for *i**. IEEE RCIS 2008, 237–246 (2008)
7. Cares, C.: From the *i** Diversity to a Common Interoperability Framework. PhD Thesis, Polytechnic University of Barcelona, Spain (2012)
8. Wachsmuth, G.: Metamodel Adaptation and Model Co-adaptation. In: Ernst, E. (ed.) ECOOP 2007. LNCS, vol. 4609, pp. 600–624. Springer, Heidelberg (2007)
9. Cares, C., Franch, X., Perini, A., Susi, A.: Towards *i** Interoperability using iStarML. *Computer Standards and Interfaces* 33, 69–79 (2010)
10. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Phd Thesis, University of Twente, The Netherlands (2005)
11. Guizzardi, R., Guizzardi, G.: Ontology-based Transformation Framework from Tropos to AORML. In: Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J. (eds.) Social Modeling for Requirements Engineering, pp. 547–570. MIT Press, Cambridge (2011)
12. Guizzardi, R., Franch, X., Guizzardi, G.: Applying a Foundational Ontology to Analyze Means-End Links in the *i** Framework. In: 6th IEEE International Conference on Research Challenges in Information Science, pp. 333–343. IEEE Press (2012)
13. Cares, C., Franch, X.: A Metamodelling Approach for *i** Model Translations. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 337–351. Springer, Heidelberg (2011)

Applying the Principles of an Ontology-Based Approach to a Conceptual Schema of Human Genome

Ana M^a Martínez Ferrandis¹, Oscar Pastor López¹, and Giancarlo Guizzardi²

¹ Departamento de Sistemas Informáticos y Computación,
Universitat Politècnica de València, Spain

{amartinez, opastor}@dsic.upv.es

² Ontology and Conceptual Modeling Research Group,
Federal University of Espírito Santo, Brazil
gguizzardi@inf.ufes.br

Abstract. Understanding the Human Genome is currently a significant challenge. Having a Conceptual Schema of Human Genome (CSHG) is in this context a first step to link a sound Information Systems Design approach with Bioinformatics. But this is not enough. The use of an adequate ontological commitment is essential to fix the real-world semantics of the analyzed domain. Starting from a concrete proposal for CSHG, the main goal of this paper is to apply the principles of a foundational ontology, as it is UFO, to make explicit the ontological commitments underlying the concepts represented in the Conceptual Schema. As demonstrated in the paper, this ontological analysis is also able to highlight some conceptual drawbacks present in the initial version of the CSHG.

Keywords: Ontology, UFO, Conceptual Modeling, Information Systems, Bioinformatics.

1 Introduction

Genomics is one of the most interesting research areas of the Bioinformatics field. Understanding the Human Genome is currently a significant research challenge but with far reaching implications such as to provide answers to questions like what the concepts that explain ours essential characteristics as species are, or how to prevent disease within a personalized medicine context. Given the large amount of data involved in such as task, and the need to structure, store and manage this data correctly, the application of sound conceptual modeling principles is made necessary. In fact, the most remarkable properties of the genomic field research are the tremendous quantity of data available, its dispersion and the continuous evolution of the involved concepts. Thus, having a Conceptual Schema of the Human Genome (CSHG) is in this context a first step to link a sound Information Systems Design approach with Bioinformatics. Moreover, given the complexity of the involved notions as well the clear need for autonomous data interoperability, it is essential that these notions are well understood and that their underlying real-world semantics are made explicit.

In this paper, we start from a concrete proposal for a CSHG [1] and illustrate how a foundational ontology (UFO) [2] can be used to make explicit the ontological commitments underlying the concepts that are represented in the Conceptual Schema. The benefits of such an approach are twofold. On one side, we can improve consistency and understandability of the CSHG through *conceptual clarification*. On the other side, this approach can identify a number of conceptual drawbacks present in the initial version of the quoted CSHG that have been put in evidence and corrected.

In the field of Biology, ontologies are often used as repositories of data, vocabularies, taxonomies, etc. A well-known, relevant example is the Gene Ontology [3]. Nevertheless, in contrast with the Gene Ontology, we here strongly advocate the use of foundational ontologies to characterize the real-world semantics that are used in the specification of conceptual genomic models. In this spirit, the work presented here is very much in line with approaches such as in [4], which promote the use of foundational ontologies to avoid errors in the curation and creation of domain models in the biomedical field. However, we here take one step forward from a conceptual modeling point of view, namely, we show how the benefits of using these foundational theories can be systematically carried out to conceptual modeling by employing an ontologically well-founded conceptual modeling language (OntoUML) [2].

The remainder of this article is organized as follows. In Section 2, we briefly present the foundation ontology UFO and its relation to the OntoUML conceptual modeling language. Section 3 explains the use of Conceptual Models in the specification of the Genomics Domain, starting with the Conceptual Schema of the Human Genome (CSHG). In Section 4, we present the main contribution of this paper, namely, the ontological analysis of the CSHG using the approach introduced in section 2. Section 5 presents some final considerations.

2 OntoUML as Tool for an Ontological Analysis of the CSHG

In recent years, there has been a growing interest in the application of Foundational Ontologies, i.e., formal ontological theories in the philosophical sense, for providing real-world semantics for conceptual modeling languages, and theoretically sound foundations and methodological guidelines for evaluating and improving the individual models produced using these languages. OntoUML [2] is an example of a conceptual modeling language whose metamodel has been designed to comply with the ontological distinctions and axiomatic theories put forth by a theoretically well-grounded Foundational Ontology [5]. This language has been successfully employed in a number of projects in several different domains including Heart Electrophysiology, Petroleum and Gas, Software Engineering, News Information Management, among many others. Asides from the language itself defined with an explicit metamodel embedded with ontological constraints, the OntoUML approach includes a number of ontology-based patterns and anti-patterns (modeling patterns, analysis patterns, transformation patterns and validation anti-patterns) as well as a number of automated tools for model construction, verification, validation, verbalization and code generation.

In section 4, we introduce some of the OntoUML modeling constructs and briefly elaborate on their ontological semantics as defined in UFO. For a fuller presentation of UFO and OntoUML, containing philosophical justification, empirical support and formal characterization, one should refer to [2,5]. We focus the remainder of this paper to illustrate with a preliminary practical exercise how OntoUML can be used to support an ontological analysis of a particular Conceptual Schema of the Human Genome, making explicit its ontological commitments, fixing a particular real-world semantics for its constructs as well as identifying conceptual problems in terms of uncertainty, inconsistencies, lack of constraints and dubious modeling choices.

3 Conceptual Schema of Human Genome (CSHG)

This section briefly elaborates on the second fundamental component for the analysis presented in this paper, namely, the Conceptual Schema of the Human Genome (CSHG) [1]. This conceptual schema was produced as a result of the Human Genome project developed by Genome Research Group of the "Centro de Investigación en Métodos de Producción de Software (ProS") of the Universitat Politècnica de València. This group is an interdisciplinary group consisting of experts both in the field of genomics and computer science whose main goal is to clearly specify and represent the genomic domain.

The CSHG consists in four different views, namely, the Variation View, the Phenotypic View, the Transcription View, and the Genome View. In the present article, due to space limitations, we focus on an excerpt of the variation view. This view comprises the description of the variations that are found on a gene. Details about this Variation View can be found in [6]. Hereafter, only the needed fragments of the CSHG that were required to understand the analyzed concepts are shown.

4 Discussion and Results

In this section we elaborate on some of the outcomes of our analysis. Due to space limitations, in this section, we restrict our discussion to fragments of the redesigned CSHG. A fuller presentation of the complete ontological analysis and redesign of the original conceptual schema will be presented in a subsequent publication. It is important to also highlight the fact that the generated OntoUML model makes explicit the particular ontological commitments underlying the CSHG as conceived by its creators. Although these particular commitments can be debated, the reason they can be so is exactly because they no longer remain tacit in the creator's minds.

The human genome is the entire genetic information that a particular individual organism has and that encodes it. It is formed by the set of all the chromosomes on the DNA. In the same manner, chromosomes are formed by a set of genes, which is an ordered sequence of nucleotides in the DNA molecule and contains the information needed for the synthesis of a macromolecule with specific cellular function.

OntoUML makes a fundamental distinction between three different types of substantial entities depending on their unity criteria and the relation they have with their

parts. Here, we focus on two of these distinctions, namely, Functional Complexes and Collectives [2,7]. A collective is an entity characterized by the fact that all its constituent parts instantiate the same type and play the same role w.r.t. the whole (e.g., a forest or a crowd). In contrast, the different parts of a functional complex X are of different types and play different roles w.r.t. to X. Examples of the latter include a human body, a computer, an organization, a TV set. In OntoUML, identity-providing rigid types whose instances are collectives receive the homonymous stereotype; identity-providing rigid types whose instances are functional complexes are stereotyped with the word «kind».

When analyzing the core concepts of the CSHG in light of these distinctions, we can see that allele, gene and chromosome can be seen as collectives and nucleotide as a functional complex (Fig. 1). These distinctions between types used in OntoUML make explicit additional information about the nature of each type. This, in turn, prevents an unwarranted interpretation that nucleotide and gene are of the same ontological nature. By making explicit the ontological nature of the entities, we can also systematically make explicit the different types of parthood relations involving these entities and their respective parts. In Fig. 1b, we have that nucleotides are essential parts of a specific Allele, i.e., besides the relation of parthood, there is an existential dependence relation between an Allele and each of its constituent nucleotides. In other words, a specific Allele can only exist (preserving the same identity) by having each of these nucleotides as parts. In fact, the identity of an allele is defined by the sum and position (sequence) of its parts.

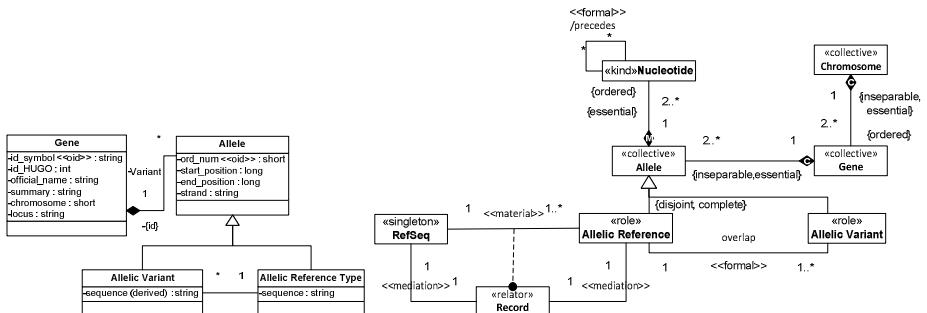


Fig. 1. (a-left) A fragment of the CSHG and (b-right) its counterpart in OntoUML (core concepts of the variation view)

In the previous version of the CSHG (Fig. 1a), the notion of nucleotide was not included. Instead, the attribute `sequence` was used to express this idea of collective of nucleotides. The existential dependence from an allele to a set of nucleotides represents explicitly the relation between the identity criteria of an allele and the ordered sequence of its constituent nucleotides. In the original model, this identity criterion was artificially represented by an assigned object identifier.

Making explicit the different types of entities in the OntoUML model clearly sets out the different types of part-whole relations involving them. OntoUML prescribes

four different types of part-whole relations: *subQuantityOf* (defined between quantities), *memberOf* (defined by individuals and the collectives they compose), *subCollectiveOf* (defined between collectives) and *componentOf* (defined between functional complexes and their parts). As demonstrated in [2], each of these different types of part-whole relations is correlated with different types of meta-properties regarding existential dependence, transitivity, shareability, among others.

Still regarding part-whole relations, Fig. 1b models that there is a mutual existential dependence between a Gene and its constituent Alleles, i.e., an Allele must be part of particular Gene and a Gene must be composed of that specific set of Alleles. Analogously, an individual gene must be part of a specific chromosome and a chromosome must be composed of that specific set of genes in every situation that it exists. Finally, in OntoUML, we have that a *memberOf* relation is never transitive, but also that *subCollectiveOf* relations are transitive [7, 8]. For this reason, in the model of Fig. 1b, we have that ultimately a chromosome can be seen as an ordered sequence of nucleotides. However, we also have that none of the parts of a nucleotide are parts of an Allele, of a Gene or of a Chromosome. In the previous version of the CSHG, all the aforementioned information remained tacit in the modeler's mind.

Alleles are specialized as *Allelic Variant* and *Allelic Reference Type*. The last is an allele that works as a stable foundation for reporting mutations, in the sense that all the alleles that are different from it (but still related to the same gene) would be classified as *Allelic Variant* and those differences would be reported as genetic variations. But what makes an allele be considered an allele of reference? The RefSeq project [9] defines the alleles to be used as standards for well-characterized genes. So, an allele becomes an *Allelic Reference* if there is a record in RefSeq for this allele. Indirectly, this record also makes the remaining alleles from a gene an *Allelic Variant*. In OntoUML, both *Allelic Reference* and *Allelic Variant* are considered types of *Role*. A role is an anti-rigid type (i.e., a type describing contingent properties of its instances) and a relationally dependent one (i.e., a type defined in terms of a relational condition) [2]. In Fig. 1, *Allelic Reference* is a role (contingently) played by an allele when referred by (*related to*) a record in RefSeq. Moreover, an *Allelic Variant* is a role played by an allele when related to the same gene as an *Allelic Reference*. Finally, an entity like *record* in Fig. 1 is modeled in OntoUML by using the notion of a *relator*. A relator is the objectification of a relational property and represents the so-called *truthmaker* of a material relation [2]. So, for instance, in the same way that an entity such as a particular marriage (a particular bundle of commitments and claims) is the truthmaker of the relation *is-married-to* between the individuals John and Mary, the presence of a RefSeq record represents here a binding between RefSeq and a particular allele, thus, making true the relation between an allele playing the role of *Allelic Reference* and RefSeq. The relations of mediation between the presence of a RefSeq record, RefSeq and the corresponding Allelic Reference (Fig. 2) are relations of existential dependence (the presence of this record depends on RefSeq and on the Allelic Reference) and constitutes the aforementioned binding.

A genetic variation is described as a difference between an *Allelic Variant* and its *Allelic Reference*. So, variations are *fiat entities* but which are existentially dependent on a particular allelic reference and a particular allelic variant. As referable entities

which are existentially dependent on multiple entities, a variation is also represented here as a relator (Fig. 2). In other words, a variation is constituted by a number of nucleotides which are part of the *Allelic Variant* and which vary *in relation to* the *Allelic Reference*. This analysis reveals a conceptual mistake in the previous version of the CSHG: since the sequence of the *Allelic Variant* is modeled as derived by the application of the variations that relate it with its *Allelic Reference Type*, the *Allelic Variant* would be characterized as existentially dependent on the variations and not the other way around. Notice that, since all parts of an allele are essential to it, we have that an *Allelic Variant* is indeed also existentially dependent on the nucleotides that constitute a variant. The notion of variant itself, however, is a relational notion that depends on both the Allelic Reference and Allelic Variant.

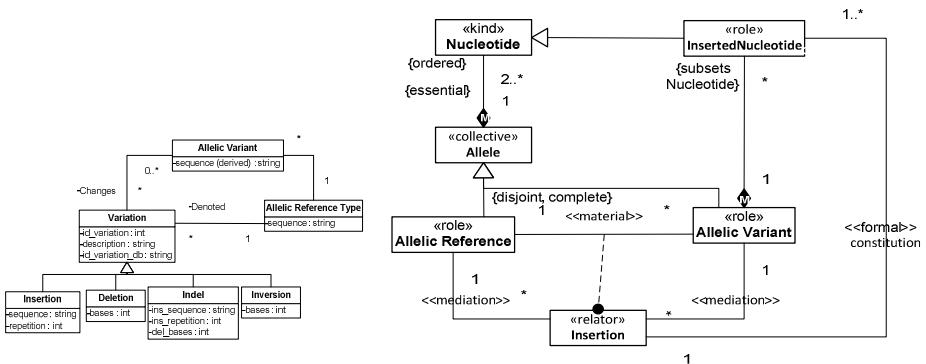


Fig. 2. (a-left) A fragment of the CSHG and (b-right) its counterpart in OntoUML (insertion as a type of variation)

Genetic variations can be further characterized depending on their type: insertions, deletions, indels and inversions. In the CSHG, they are simply represented as subtypes of the *Variation* class. This incompleteness in the model, however, leaves implicit the fact that each of these variations is derived from different types of *base relations*. In the redesigned model, we use the OntoUML relator construct to represent explicitly each category of variation with its characterizing relations. The example that we show in Fig. 2 is the case of *insertions*. In the case of insertions, we have that the nucleotides that *constitute* an insertion are parts of the allelic variant mediated by this variation (i.e., of which this variation depends). In the model of Fig. 2, these nucleotides are said to play the role of *Inserted Nucleotides* w.r.t. the allelic variant. Moreover, the part of relation between the former and the latter is explicitly represented in that model. Since an *Allelic Variant* is a collective, this is an example of a *memberOf* relation [7, 8]. This model should also include a constraint that the nucleotides that play the role of *Inserted Nucleotides* w.r.t. an *Allelic Variant* must *constitute* the insertion which mediates that Allelic Variant. Moreover, the nucleotides which play the role of *Inserted Nucleotides* in an insertion and which are members of an *Allelic Variant* are necessarily member of that specific allele playing the role of *Allelic Variant*. This inclusion constraint is represented via association

subsetting in Fig. 2b. The need for this constraint can be automatically detected in an OntoUML model since its absence would include in the model an instance of a pre-defined validation OntoUML anti-pattern [10].

Finally, in the human genome, there is also the notion of *conservative regions*, which are regions that have been in the genome for ages without alteration and which are expected to remain the same in the allelic reference and its variants. We use here a formal relation from the mereological theories underlying OntoUML to model that there exists a relation of (*non-proper*) *overlapping* between an *Allelic Reference* and its *Allelic Variant* (Fig. 1 and Fig. 2). In other words, Allelic Reference and Allelic Variant must share a common part. If there is no overlapping between sequences, then the two alleles belong to different genes. This constraint is of significance when talking about the nature of the alleles and genes, another feature which remains implicit in the previous version of the CSHG.

5 Conclusion

In this paper, we start from a concrete proposal for a Conceptual Schema for the Human Genome and illustrate how a principled ontological analysis can be used to make explicit the ontological commitments underlying the concepts that are represented in that schema. Moreover, the paper illustrates an approach in which this ontological analysis is performed systematically and is integrated in a classical conceptual modeling engineering activity throughout the use of an ontologically well-founded conceptual modeling language termed OntoUML as well as its associated methodological tools.

In the redesign of the CSHG as an OntoUML model, a number of implicit assumptions in the original model were made explicit as well as a number of conceptual drawbacks were identified. For instance, the introduction of the RefSeq and the Record was instrumental for expressing that Allelic Reference and Allelic Variant are contingent roles played by an allele in relational contexts: in order for an allele to be an allelic reference it must be referred by a RefSeq record; in order for an allele to be an allelic variant it must be related to the same gene and non-properly overlap with an allelic reference. Moreover, the use of the formal relation of non-proper overlapping between an Allelic Reference and its Allelic Variant represents the conservative regions on the DNA, making explicit the constraint of the non-existence of “extreme variations” in the domain. Furthermore, modeling the Variation as a relator also expresses its existential dependency on a specific Allelic Variant and on a specific Allelic Reference. This highlights the doubtful choice of considering the sequence of the Allelic Variant as derived by the application of the variations that relate it with its Allelic Reference Type. Finally, the use of the mereological relations of memberOf and subcollectiveOf to represent parthood between concepts such as Gene, Nucleotide, Allele and Chromosome makes explicit the notion of a chromosome as an ordered sequence of nucleotides.

The analysis presented here concentrates on a small fragment of the Variation view of CSHG. In a future work, we shall present a full analysis of CSHG contemplating

all its constituent views. Once we have a complete OntoUML version of the CSHG, we pretend to conduct a full validation with domain experts by using the OntoUML approach of model validation via visual simulation [10]. Finally, after validation, we intend to use the OntoUML tool set to automatically generate OWL specifications for the CSHG. These specifications, in turn, will be employed to support semantic annotation and automated reasoning in a Human Genome Wiki environment.

Acknowledgements: This work has been developed with the support of MICINN under the project PROS-Req TIN2010-19130-C02-02 and the Programa de Apoyo a la Investigación y Desarrollo (PAID-00-12) de la Universitat Politècnica de València, and co-financed with ERDF. The third author has been supported by FAPES (PRONEX Grant #52272362/2011).

References

1. Pastor, O., Levin, A.M., Casamayor, J.C., Celma, M., Eraso, L.E., Villanueva, M.J., Perez-Alonso, M.: Enforcing conceptual modeling to improve the understanding of human genome. In: 2010 Fourth International Conference on Research Challenges in Information Science (RCIS), pp. 85–92. IEEE (2010)
2. Guizzardi, G.: Ontological foundations for structural conceptual models. CTIT, Centre for Telematics and Information Technology (2005)
3. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T.: Gene Ontology: tool for the unification of biology. *Nature Genetics* 25, 25–29 (2000)
4. Burek, P., Hoehndorf, R., Loebe, F., Visagie, J., Herre, H., Kelso, J.: A top-level ontology of functions and its application in the Open Biomedical Ontologies. *Bioinformatics* 22, e66–e73 (2006)
5. Guizzardi, G., Wagner, G.: Using the Unified Foundational Ontology (UFO) as a foundation for general conceptual modeling languages. In: Theory and Applications of Ontology: Computer Applications, pp. 175–196. Springer (2010)
6. Pastor, O., Levin, A.M., Casamayor, J.C., Celma, M., Kroon, M.: A Conceptual Modeling Approach to Improve Human Genome Understanding. In: Handbook of Conceptual Modeling, pp. 517–541. Springer (2011)
7. Guizzardi, G.: Ontological foundations for conceptual part-whole relations: the case of collectives and their parts. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 138–153. Springer, Heidelberg (2011)
8. Guizzardi, G.: Representing collectives and their members in UML conceptual models: an ontological analysis. In: Trujillo, J., et al. (eds.) ER 2010. LNCS, vol. 6413, pp. 265–274. Springer, Heidelberg (2010)
9. QPruitt, K.D., Tatusova, T., Maglott, D.R.: NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research* 35, D61–D65 (2007)
10. Sales, T.P., Barcelos, P.P.F., Guizzardi, G.: Identification of Semantic Anti-Patterns in Ontology-Driven Conceptual Modeling via Visual Simulation. In: 4th International Workshop on Ontology-Driven Information Systems (ODISE 2012), Graz, Austria (2012)

Ontologies for International Standards for Software Engineering

Brian Henderson-Sellers¹, Tom McBride¹, Graham Low², and Cesar Gonzalez-Perez³

¹ Faculty of Engineering and Information Technology, University of Technology,
Sydney, P.O. Box 123, Broadway, NSW 2007, Australia

² School of Information Systems, Technology and Management,
University of New South Wales, Sydney, NSW 2052, Australia

³ Institute of Heritage Sciences (Incipit)

Spanish National Research Council (CSIC), Santiago de Compostela, Spain

{brian.henderson-sellers, tom.mcbride}@uts.edu.au,
g.low@unsw.edu.au, cesar.gonzalez-perez@incipit.csic.es}

Abstract. International Standards have often been developed independently of one another resulting in the multiple use of similar terminology but with different semantics as well as the more obvious dependencies between pairs of standards employing the same term with the same semantics. By the application of conceptual modelling techniques based on an ontological viewpoint, we show how the ‘stovepipes’ of software engineering standards, developed under the remit of the SC7 committee of ISO, can be reconciled into a single coherent suite of standards.

Keywords: conceptual modelling, international standards, ontologies.

1 Introduction

The need for International Standards published by ISO (International Organization for Standardization, based in Geneva) to be consistent with each other in terms of terminology, structure and semantics has long been recognized and debated. For example, within the SC7 (the sub-committee responsible for software engineering standards) community, Rout [1] analyzed a number of standards, term by term, identifying the standards in which terms appear together with their (disparate) definitions.

In the early 2000s, SC7 standards were, however, being developed increasingly in ‘stovepipes’ by individual working groups within SC7, which led to significant discussions within (especially) Special Working Group 5 (hereafter SWG5) as well as the initiation of a formal goal of harmonizing two specific and overlapping process-focussed standards: ISO/IEC 12207 and ISO/IEC 15288, both within the remit of Working Group 7 of SC7 [2].

At the beginning of 2012, McBride et al. [3] produced a document for the 2012 SC7 Plenary meeting (Korea) entitled “The growing need for alignment”, which suggested, inter alia, a need to move from serendipitous knowledge of such problems to organizational (SC7-level) solutions and a need for an underpinning ontology.

Resulting from the discussion of these proposals at the May 2012 meeting of SC7, a Study Group was created, chaired by the first author of this paper and charged with investigating the potential utility of ontologies for rationalizing SC7's suite of software engineering International Standards.

In this paper, we present the research-oriented results of the material evaluated by the members of this Study Group and the proposal that was made to the SC7 overseeing committee (SWG5) at the SC7 Plenary meeting in Montréal in May 2013 and the subsequent revisions to create a "Second Report" (circulated to committee members July 2013). At that May 2013 plenary meeting, further work was also commissioned by SWG5 in order to create a work plan prior to the development of the report into a New Work Item Proposal (NWIP) at the next international SC7 meeting. Our overall hypothesis is simply that SC7 standards, as an interdependent suite of artefacts, could benefit by the application of conceptual modelling and ontological organization.

In summary, the problems to be addressed in this paper and the proposed solutions adopted by the Study Group and SWG5 are: (i) construction of clear, unambiguous and comprehensive definitions of all SC7 terminology; (ii) conformance of existing and new standards to this agreed ontological description of terminology; (iii) categorization of existing standards and their relationship to (i) and (ii).

The approach taken is that of a combination of conceptual modelling e.g. [4] and ontology engineering e.g. [5]. In Section 2 of this paper, we first identify from the research literature various flavours of 'ontology' that might be useful for SC7. Section 3 discusses how these ideas might be implemented in the context of SWG5's request in May 2013 for a detailed work plan. Section 4 discusses the reality of SC7 adoption of this proposal now and in the future, whilst Section 5 gives both the recommendations (that will be made to SC7) together with our overall conclusions.

2 Proposals: An Ontological Underpinning to SC7 Standards

We first identified five distinct areas where conceptual modelling and ontologies might be helpful for reorganizing SC7 standards: three foundational or upper-level ontologies [6, 7] (discussed in Sections 2.1, 2.2 and 2.5) and two domain ontologies (Sections 2.3 and 2.4).

2.1 Definitional Elements Ontology (DEO)

The proposal for the Definitional Elements Ontology (DEO) is at the core of this newly proposed infrastructure. Whilst not an ontology in the strictest sense, the DEO provides definitions of individual concepts/terms, together with relationships between these concepts. Each definition (concept) adheres to a pre-determined template, which includes a name, a definition, and a set of properties. The concepts in the DEO will be taken from a combination of several existing ISO sources: (1) ISO/IEC 24765 (SEVCAB) is a collection of terms from SC7 standards that are frequently duplicated but slightly different in expression or meaning and has been identified as a valuable input to creating the standardized DEO together with (2) ISO/IEC 24744 (SEMDM), which will be used as the basis for an ontology for software engineering [8].

These two major sources will be augmented with concepts from additional sources. Since SEVOCAB already takes into account concepts (and definitions) from a large number of standards and technical reports, it is only necessary to examine those additional sources not explicitly treated by SEVOCAB: viz. ISO/IEC 15437, 18152, 18529, 24773, 24774, 29148 and 42010; plus 11 standards still in development. Concepts are elicited from those elements in the “Terminology” section of these standards. Additional input is from the 69 concepts in ISO/IEC 24744 SEMDM, for which there might be up to 7 corresponding definitions in ISO/IEC 24765 SEVOCAB. Since rationalizing these multiple definitions could take, on average, several hours for each term, the workplan (to be delivered in November 2013) will need to include an overall estimate of the likely effort involved in implementing the research ideas described here.

Necessarily, the DEO is highly abstract, in order to guarantee that it covers everything under the scope of SC7. Therefore, the DEO cannot be utilized without change for all standards. Rather, it needs to be tailored to the specific needs of each working group or standard.

The research methodology is thus to identify core concepts from these two main sources (ISO/IEC 24765 and 24744) and to abstract away to a core definition from which all the other definitions can be derived in their individual CDOs.

2.2 Configured Definitional Ontology (CDO)

Although the elements in the DEO are well connected, zero-to-many cardinalities on certain relationships permit ‘pruning’ of the DEO when it is necessary to create a tailored and more specific standard. Such tailored standards may be the set of standards within a single working group that therefore all use the same configuration (CDO) or within a working group as a “family” of standards (see discussion below) or at the individual standard level.

A ‘Configured Definitional Ontology’ is a specific configuration of the elements in the DEO – either as a standalone standard (*à la* ISO/IEC 24744) or as an intermediate template for the creation of a ‘Standard Domain Ontology’ a.k.a. an International Standard (IS) (Section 2.3). A CDO gives structure and order to the concepts used and necessary in a specific domain. For example, ISO/IEC 15504 needs to define a process in terms of its purpose and outcomes. Without those two concepts, process assessment could not be consistent or rigorous. It happens that processes within SC7 are comprised of more than their purpose and outcomes and, as the impact of other domains become evident within SC7, a larger CDO becomes more important. An example of such an ‘other domain’ is that of governance. The requirements for auditable governance will emerge over time and these, like that of quality, will introduce new activities, new work products, and possibly new relationships between different processes. All of these may throw up new concepts that need to be evaluated for possible addition to the DEO.

A CDO is created by using a combination of the two following mechanisms: (i) Discarding areas of the DEO that are not relevant; and (ii) Adding new concepts, as refinements of those in the DEO. These two mechanisms are illustrated in Fig. 1 and described in detailed in the two subsections below.

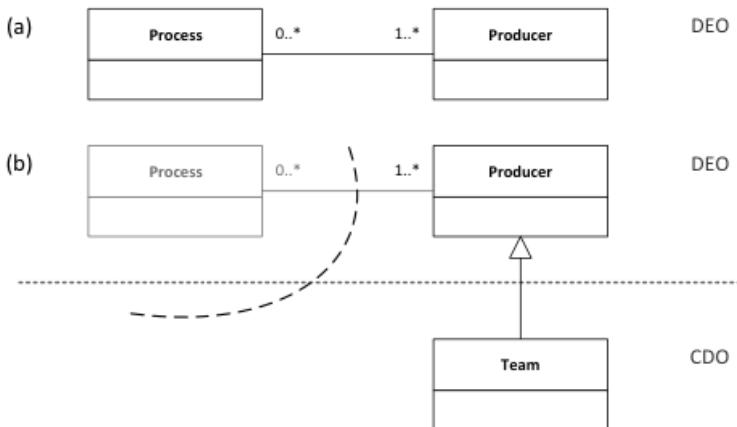


Fig. 1. Mechanisms for tailoring the DEO: (a) shows the DEO in its original form. (b) shows the DEO as being used by a particular CDO: a section of the DEO is being discarded and a subclass has been added.

Discarding Areas of the DEO. Although the elements in the DEO are all interconnected, associations between two concepts may have a cardinality of zero; this means that for any occurrence of the concept on one side of the association, there may be multiple or no occurrences on the opposite side. For example, Fig. 1 shows concepts Process and Producer plus an association between them with a zero-to-many cardinality on the Process side. This means that any particular producer (such as software developer Jane) may have no processes at all. For this reason, a project focussing on modelling producers could legitimately discard the Process concept altogether and use only the Producer concept. However, a project focussing on processes could not discard the Producer concept, since every possible process occurrence would need at least an associated producer.

Adding Elements to the DEO. As well as discarding unwanted sections of the DEO, new concepts can be introduced into a CDO. This is often the case when specialized definitions are needed that are not present in the abstract DEO.

Newly added concepts are introduced by refining an existing concept in the DEO through specialization. This means that the newly introduced concept must be semantically “a kind of” the base concept. For example, Fig. 1 shows Team as a specialized concept that has been introduced as a kind of Producer.

Chaining CDOs. CDOs are always created in a particular context, which determines the relevant set of concepts. This context may be a working group, a family of standards, or even a particular standard. Thus the DEO can be tailored to two different SC7 working groups by creating two CDOs through the mechanisms described above. In turn, each of these CDOs can be further refined into a suite of CDOs that are specific for families of standards within that working group. These are then refined into standard-specific CDOs.

In the chain of CDOs, detail is added gradually so that only the concepts that are relevant to a particular context (either a working group, a family of standards or a single standard) are included. Care must be taken so that under- or over-specification is avoided. This chaining mechanism ensures that each working group retains ownership of its own set of concepts, while (a) maintaining the interoperability with other working groups through the shared DEO and (b) permitting individual standards to refine these concepts into more specific ones.

2.3 Standard Domain Ontology (SDO)

A ‘Standard Domain Ontology’ (SDO) is created by instantiating a CDO i.e. it contains a number of instances conforming to the CDO template. An SDO becomes an International Standard (or a Technical Report (TR)). It is important to discriminate between the definitions (the DEO and the CDO: foundational ontologies) as compared to examples (instances) of those definitions – the SDO: a domain ontology [6].

An SDO is highly specific to one particular situation e.g. process management, services, governance. For example, in a process-focussed standard such as ISO/IEC 12207, it may only be necessary to utilize definitions for, say, Process, Activity, Task and their inter-relationships, which provide the template for the creation of instances of these definitions that are then put together as a CDO, wherein each 12207 ‘process’ can be assessed for conformance to the definition of Process given in the DEO . In other words, a check that all the elements of the new IS or TR (an SDO) are predetermined by the definition of ‘Process’ in the CDO, this definition having been selected from all the elements in the DEO.

2.4 Standards Relationship Diagram (SRO)

Originally, it was proposed to create a Standards Relationship Diagram (SRO) as a domain ontology of these produced standards. However, in the SWG5 discussions after the May 2013 SC7 Plenary meeting, it was realized that such an SRO could more usefully be regarded as an input, not an output, to this project since with the revised DEO/CDO structure, influence connections are “upward” via the DEO, not “sideways” because of shared concepts in their Terminology section.

2.5 Advanced Foundational Ontology for Standards (AFOS)

The fifth original proposal (deferred in the revisions submission to SC7) was for an ‘Advanced Foundational Ontology for Standards’ (AFOS): an extended formalism to the DEO in terms of standard concepts from the ontological engineering literature, such as sortals and moments [9,10]. These concepts are important in the context of work products and, in particular, modelling languages (SC7 WG19’s remit). For example, future versions of UML (ISO/IEC 19505) and UML for RMODP (ISO/IEC 19793) are two standards that will be impacted.

3 Implementation and Initial Validation

After receiving feedback in May 2013 following presentation to SWG5 and other working groups at the SC7 Plenary in Montréal, it was decided that a better structure would be to focus on the DEO and the derived CDOs, as described above. SDOs are then identified simply as being the suite of International Standards conformant to the CDOs.

The first implementation step is to identify and agree upon definitions of a wide variety of concepts used in software engineering. These constitute the DEO. As noted above, the DEO definitions will need to be collected from various sources: 24744 and 24765, supplemented by those ISO standards not contributing to ISOIEC 24765.

The importance of having a DEO applicable to all SC7 standards can be demonstrated by consideration of what happens when a NWI proposes a standard in a ‘new’ domain. Advocates of this new domain standard are likely to have different tacit understanding of the terms they use. It is thus critical that a first step is to evaluate the definitions (semantics) of all terminology in the proposed new domain and to ensure it is compatible with the SC7 DEO. In some circumstances, for example when brand new technologies arise, there may be a need to add extra elements to the DEO but this should be done carefully to ensure they are well defined and there are no clashes (contradictions or overlaps) with existing DEO definitions. In other cases, synonym mappings may be usefully identified in order to retain consistency in both the domain-specific standard (thus Working Groups retain ‘ownership’) and also the coherence of the elements in the DEO.

Another important benefit of the proposed approach is that, as explained above and thanks to the CDO chaining mechanism, each working group continues to own the concepts and terms that it introduces but with the obligation that these concepts and terms conform to (either directly or via refinement) concepts in the DEO. Once a concept is in the DEO, a working group can use synonyms of the terms or refinements of the concepts in their work as they see fit.

It is an important recommendation that the route of a single comprehensive DEO MUST be taken, thus avoiding the use of multiple CDOs, since that mimics the diversity (and incompatibilities) identified in the current SC7 standards.

4 Maintenance, Responsibilities and Future Adoptions

Once the DEO has been standardized as either an International Standard or a Technical Report within SC7, three processes begin in parallel: (1) The DEO is maintained on a continuous basis; (2) Existing standards get reconciled against the DEO; and (3) New standards will need to be aligned to the DEO, as described in Section 4.3.

4.1 Maintaining the DEO

Although the DEO will be a standard and therefore should be relatively stable over a period of years, it is inevitable that, eventually, changes will need to be introduced. It is recommended that SWG5 take responsibility for maintaining the DEO, incorporating the changes that they see fit from feedback obtained, either proactively or reactively, from the different working groups in SC7.

4.2 Reconciling Existing Standards

We are aware that altering existing standards to conform to the DEO may entail a significant amount of work in some cases. It is recommended that existing SC7 standards are considered for DEO reconciliation whenever they become open for periodical review. The owner working group should be responsible for this task.

Reconciliation entails re-describing the standard as a CDO and, if necessary, a collection of occurrences of said CDO. If the focus of standard is the occurrences of concepts (e.g. ISO/IEC 12207), then an explicit CDO is still necessary as part of the reconciled version in order to document the semantics of said occurrences.

In any case, reconciling a standard with the DEO would only mean an internal refactoring with no changes whatsoever to the semantics of its prescriptive parts. In other words, the wording and terminology describing the concepts in the reconciled version of the standard may need to be changed in order to refer to the DEO, while maintaining their original semantics. Often, it will be necessary to map the standard's terms and definitions to the SC7 DEO, revise the standard to use standard terms where possible and migrate terms to the SC7 DEO over time.

4.3 Creating New Standards

Any new standard that is created after the DEO is in place will be required to be fully compliant with the DEO. This means that the concepts in it must be a CDO. If the focus of standard is the occurrences of concepts (e.g. ISO/IEC 12207), then an explicit CDO is still necessary as part of the new standard in order to document the semantics of said occurrences. This statement will replace the content of the "Terms and definitions" section present in all current SC7 standards.

4.4 Guidance and Tools

Once all of the above has been implemented, there will be a need to provide guidance for would-be standards developers with regard to the process to follow in relation to CDO development, CDO instantiation etc. A flow chart or decision tree may be useful in order to guide potential standards builders. In addition, it is suggested that ontology management tools be employed so that relationships between standards can easily be tracked and the DEO/CDO structure be easily maintained.

5 Recommendations and Conclusion

Once the above ontological structure has been implemented, there will be a need to formalize these not only in revised standards such as ISO/IEC 12207 etc. (ensuring they conform to the elements in the DEO and to one specific CDO) but also to introduce new standards and technical reports for would-be standards developers as guidelines to their standards activities.

- A new suite of standards is thus required to explain and formalize all this.
- i) There is a need for a standard (or Technical Report) to explain how (a) to put together and (b) how to tailor such a standard.
 - ii) Because the spectrum of standards is broad and we can't change accepted terminology in all subdomains, there MUST a synonym list so that non-standard standards can relate back to the standard vocabulary.

We have shown how conceptual modelling can be efficacious in reconciling otherwise disparate International (ISO) Standards – here the software engineering standards developed under the auspices of SC7. We have applied ontological thinking to the various existing standards and identified areas in which ontological engineering can contribute immediately to increasing the quality of the SC7 suite of software engineering standards. The core is the DEO; a comprehensive network of higher-level definitions from which domain-specific CDOs can be tailored. Each CDO is then a metamodel/foundational ontology to which an individual IS or TR must conform. These recommendations will form part of a second submission to be made to the SC7 committee and its working groups as a prelude to the creation and endorsement of a NWI (New Work Item) in 2013-14, which will finally culminate in a new IS (the DEO).

References

1. Rout, T.P.: Consistency and conflict in terminology in software engineering standards. In: Procs. Fourth IEEE Int. Symposium and Forum on Software Engineering Standards (ISESS 1999), pp. 67–74. IEEE Computer Society (1999)
2. Details of all ISO standards can be found at <http://www.iso.org>
3. McBride, T., Henderson-Sellers, B.: Reviewers: The growing need for alignment, ISO/IEC JTC1/SC7 N5507 (May 12, 2012)
4. Kaschek, R., Delcambre, L. (eds.) The Evolution of Conceptual Modeling. LNCS, vol. 6520. Springer, Heidelberg (2011)
5. Guizzardi, G.: Theoretical foundations and engineering tools for building ontologies as reference conceptual models. Semantic Web 1, 3–10 (2010)
6. Henderson-Sellers, B.: Bridging metamodels and ontologies in software engineering. Journal of Systems and Software 84(2), 301–313 (2011)
7. Guizzardi, G., Wagner, G.: Towards ontological foundations for agent modelling concepts using the Unified Foundational Ontology (UFO). In: Bresciani, P., Giorgini, P., Henderson-Sellers, B., Low, G., Winikoff, M. (eds.) AOIS 2004. LNCS (LNAI), vol. 3508, pp. 110–124. Springer, Heidelberg (2005)
8. Gonzalez-Perez, C., Henderson-Sellers, B.: An ontology for software development methodologies and endeavours. In: Calero, C., Ruiz, F., Piattini, M. (eds.) Ontologies in Software Engineering and Software Technology, pp. 123–152. Springer-Verlag (2006)
9. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Enschede, The Netherlands (2005)
10. Henderson-Sellers, B.: On the Mathematics of Modelling, Metamodelling, Ontologies and Modelling Languages. Springer Briefs in Computer Science. Springer (2012)

On the Symbiosis between Enterprise Modelling and Ontology Engineering

Frederik Gailly¹, Sven Casteleyn^{2,3}, and Nadejda Alkhaldi²

¹ Ghent University, Ghent, Belgium

² Universitat Jaume I, Castellon, Spain

³ Vrije Universiteit Brussel, Brussels, Belgium

Frederik.Gailly@ugent.be,

{Nadejda.Alkhaldi,Sven.Casteleyn}@vub.ac.be

Abstract. In different fields, ontologies are increasingly deployed to specify and fix the terminology of a particular domain. In enterprise modelling, their main use lies in serving as a knowledge base for enterprise model creation. Such models, based on one or several compatible so-called enterprise-specific ontologies, allow for model alignment and solve interoperability issues. On the other hand, enterprise models may enrich the enterprise-specific ontology with concepts emerging from practical needs. In order to achieve this reciprocal advantage, we developed an ontology-based enterprise modeling meta-method that facilitates modelers to construct their models using the enterprise-specific ontology. While doing so, modelers give their feedback for ontology improvement. This feedback is subject to community approval, after which it is possibly incorporated into the ontology, thereby evolving the ontology to better fit the enterprise's needs.

Keywords: enterprise modelling, ontology engineering, ontologies.

1 Introduction

In modern day enterprise engineering, it is paramount that enterprise models are grounded in a well-defined, agreed-upon Enterprise Architecture that captures the essentials of the business, IT, and its evolution. Enterprise architectures typically contain different views (e.g. Business, Information, Process, Application, Technical) on the enterprise that are developed by distinct stakeholders with a different background and knowledge of the business. Consequently, the developed enterprise models that populate these views are hard to integrate. A possible solution for this integration problem is using a shared terminology during the development of these different views [1]. Such explicit formal representations, often materialized in the form of an ontology – in a business context called an enterprise-specific ontology - provide a myriad of advantages. On an intra-organizational level, they ensure model re-usability, compatibility and interoperability, and form an excellent basis for enterprise-supporting IT tools, such as Enterprise Resource Planning (ERP) systems, business intelligence (BI) tools or information systems (IS), for which they serve as common terminology. On an inter-organizational level,

they facilitate interoperability, cooperation and integration by allowing formal mappings between, and alignment of separately developed enterprise models. While a wide range of more generic enterprise ontologies with a various intended use are available [2–4], they are often not immediately usable by a particular enterprise, as they lack enterprise-specific business concepts which enterprise-specific ontologies do provide, or do not offer a complete coverage of the business domain. On the other hand, in ontology engineering, an important stream of research advocates the creation of ontologies as a shared, community-based effort [5]. In this view, rather than being put forward by a single authority, ontologies grow out of a community of stakeholders, and evolve to represent a shared agreement among them [6]. Nevertheless, assembling and motivating such a community proves to be a tedious task, as potential stakeholders need proper incentives to contribute.

In this article, we aim to establish a symbiotic relationship between ontology engineering and enterprise modelling by offering enterprise engineers an ontological foundation and support to create enterprise models, while at the same time leveraging this enterprise modelling effort to contribute to community-based ontology engineering. Concretely, we present a meta-method that 1/ provides the enterprise engineer with design support as he is creating the enterprise models, by offering suitable modelling elements based on the available concepts and relations in the enterprise-specific ontology, 2/ supports evaluation of the created enterprise models based on well-established evaluation criteria, 3/ utilizes the feedback originating from the enterprise modelling effort to contribute to the improvement and evolution of the enterprise-specific ontology.

Our meta-method contributes to both the enterprise modelling and ontology engineering fields. First, it eases and speeds up the creation of enterprise models by offering the modeller concrete modelling suggestions. By exploiting the knowledge available in the enterprise-specific ontology to do so, it also promotes completeness and correctness of these models. Furthermore, by grounding the enterprise models in the enterprise-specific ontology, enterprise engineers obtain the previously mentioned inter- and intra-organizational advantages. From an ontology-engineering point of view, our method increases community involvement, as enterprise engineers assist in community-based ontology evolution by suggesting concepts or relations that were found to be missing, or contrarily, found to be ill-defined or too restrictive. Over time, as community mediation plays its role, this improves the consensus about, and the completeness of the enterprise-specific ontology, which evolves to a more suitable basis for enterprise modelling for the particular enterprise.

2 Related Work

Enterprise modeling is an activity where an integrated and commonly shared model of an enterprise is created [7]. The resulting enterprise model comprises several sub-models, each representing one specific aspect of the enterprise, and each modeled using an appropriate modeling language for the task at hand. For example, the enterprise model may contain processes modeled in BPMN, data modeled in ER and goals

modeled in i*. The enterprise model is thus developed by several enterprise engineers, and aggregates all information about the enterprise. As a result, enterprise models without homogenized underlying vocabulary suffer interoperability and integration problems [8].

It is generally accepted that ontologies can be used to achieve interoperability. In the context of enterprise modelling, ontologies have been used for multiple purposes. One consists of evaluating existing modelling languages, which allows improving the definition of the modelling languages constructs and allows language developers to integrate the construct of different languages [8]. Another use is to create ontology-driven business modelling languages, which primarily resulted in new types of enterprise modelling languages that focus on a specific part, or provide a specific view, of an enterprise [3]. Another approach for solving integration and interoperability issues between enterprise models is using Enterprise Architectures [9]. Techniques like TOGAF¹ and Archimate¹ focus on describing different aspects of an enterprise in an integrated way. The actual level of integration realised between the perspectives of the enterprise architecture and the approach followed to realize depends on the used framework.

All previously described approaches for solving interoperability and integration issues between enterprise models differ from our approach. First, we focus on using an enterprise-specific ontology and not generic enterprise ontologies that are shared by different enterprise. Second, we do not impose particular modelling language(s) but instead allow the use of generally accepted languages that capture different aspects of the business. Finally, in contrast to the aforementioned approaches, the used enterprise-specific ontology is used to provide suggestions to the modeller, and evolves according to the specific needs of the enterprise.

As mentioned, the method proposed here does not only support the development of enterprise models, it also facilitates enterprise engineers to participate in community-based ontology development. An overview of different ontology engineering methods can be found in [6]. Although all these methods have as goal developing a shared ontology, it remains an open question how they will enforce this sharedness [10]. Relevant to this paper, there were previous attempts on real community involvement in the ontology engineering process [11, 12]. In all these approaches, the community participates in ontology creation, but it is left open who constitutes the community or how to provide incentive for participation. Furthermore, the ontology is created and subsequently used, but cannot evolve according to users (i.e., enterprise) needs.

In the meta-method proposed in this paper the enterprise-specific ontology is used on one hand to assist the enterprise modeller. While doing so, enterprise engineers generate feedback for the enterprise-specific ontology, which can be used by the community of enterprise stakeholders to gradually refine and evolve the ontology. This results in an ontology that gradually matches the enterprise needs better, and thus provides immediate incentive for community participation. A method that seeks to establish such a symbiosis between enterprise modelling on one hand, and ontology engineering on the other, is to the best of our knowledge unique.

¹ <http://www.opengroup.org/standards/ea>

3 Meta-Method

Figure 1 provides a general overview of our meta-method, and shows the interplay between ontology engineering on one hand, and enterprise modelling on the other hand. We consider the described method a meta-method as no specific algorithms, heuristics, ontology or modelling languages are prescribed. In other words, before the method can be used in practice, both the ontology engineering and enterprise modelling cycles first need to be instantiated.

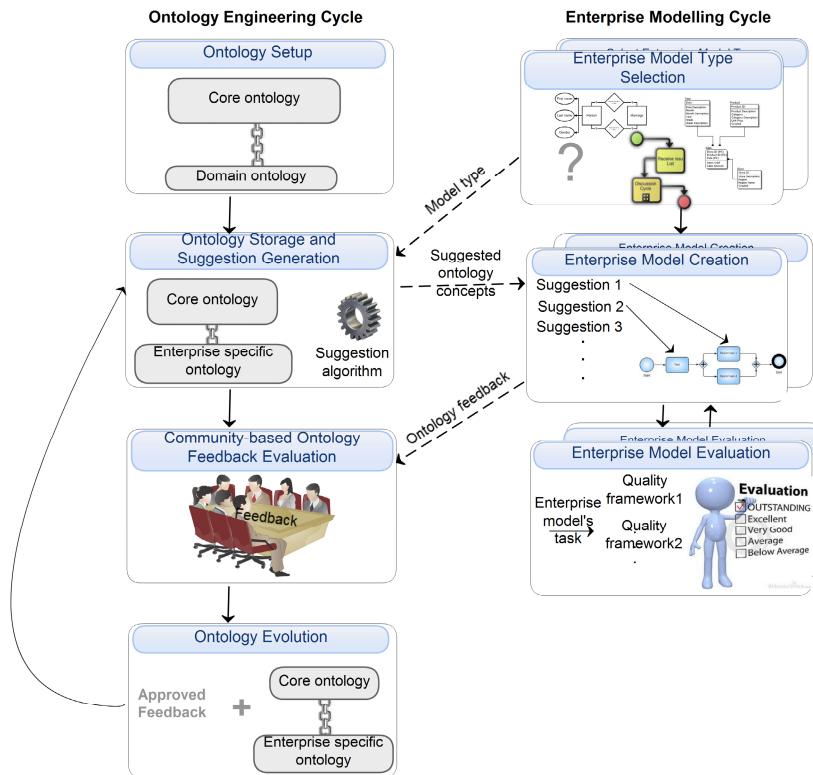


Fig. 1. Meta-Method

Before the two cycles are described in detail, we first shortly overview some relevant concepts from ontology engineering that are required to fully grasp our approach. Ontology researchers make a distinction between core ontologies, domain ontologies and application ontologies [13]. A core (high-level) ontology describes general concepts that are independent of a specific domain. A domain ontology describes the concepts, relations and axioms related to a generic domain (like medicine, or automobiles) by specializing the terms introduced in the core ontology. Finally, an application ontology adds specificities to the domain ontology which are only relevant for the application considered, but are not shared across the entire domain. In the context of

our meta-method, the application considered is in enterprise modelling. Consequently, in this case, the application ontology is called the enterprise-specific ontology, and is used to support the development of different types of enterprise models.

3.1 Ontology Engineering Cycle

The first step of the meta-method consists of setting up the enterprise-specific ontology. During the instantiation of the **Ontology Setup** phase, the enterprise needs to decide which core and/or (initial) domain ontology will be used during the development of the enterprise-specific ontology. In essence, the domain ontology is specified in terms of a core ontology (e.g., UFO, SUMO), and is used as a starting point for the enterprise-specific ontology, which evolves towards the enterprise' particular needs throughout our method. The selection of these ontologies depends on a multitude of factors. In principle, the choice of core and domain ontology is completely free. However, from an enterprise point of view, the deciding factor is often the availability of a domain ontology that is closely related to the business. Alternatively, a domain ontology can be created using the artefacts (e.g., glossaries, vocabularies, informal sources such as excel files of use case descriptions) that are already available in the enterprise. Existing domain ontologies often come in one particular representation format (e.g., OWL, UML), and may or may not be defined in terms of a core ontology. Nevertheless, from the point of view of our method, the choice of core and domain ontology may significantly reduce the amount of initialization work required. Modelling suggestions are based on well-defined connections between the domain and core ontology, and consequently, if the domain ontology is not yet defined in terms of the core ontology, this will be a major task in the ontology setup phase. On the other hand, the richness of the core ontology influences the richness of possible modelling suggestions, and is thus also an important issue to consider. Finally, the modelling suggestions are based on the relation between a core ontology and a target enterprise modelling language. For some, existing analyses are available, which eases the generation of suggestions. For instance the UFO core ontology has been used to analyse UML class diagram modelling [14]. If no previous analysis is available, the suggestions depend on custom mappings defined in our method, which do not benefit of equally rigorous scientific research, and may thus be of lower quality.

Once the enterprise-specific ontology has been set up, it is subsequently used to **generate modelling suggestions**. The suggestion creation algorithm uses three mechanisms for providing suggestions to the modeller: a search engine, core ontology matching mechanisms and contextual visualisation mechanisms. First, the modeller selects a modelling construct to add to his model. Next, the modeller enters the name for the new modelling construct, after which the search engine uses string-matching algorithms to search for matching concepts in the enterprise-specific ontology, and ranks the results. Finally the enterprise-specific concepts that are defined in the ontology also contain contextual information that is visualized in a condensed way to provide the modeller with useful information to assist him in identifying the best-suited ontology concept. Examples of such contextual information include synonyms, relationships between concepts, notes or classifications.

Prompted by completion of an enterprise modelling effort, the Ontology Engineering cycle initiates an **Ontology feedback Evaluation phase**. During this phase the

feedback given during a model development cycle is made available to the community members and is subject to discussion, until a final consensus is reached whether or not the proposed change(s) should be included in an evolved version of the ontology. Community members are all enterprise stakeholders who are involved in the enterprise modelling efforts. According to our meta-method, any community-based consensus method can be used. For instance, in case the community members are not co-located, both in time and space, and as we are aiming to progressively reach a consensus, the Delphi approach [15] is an excellent candidate. This approach is perfectly suited to capture collective knowledge and experience of experts in a given field, independently of their location, and to reach a final conclusion by consensus. Only in case the community is not able to reach a consensus, the final decision is made by community members with a high level of trust or by an authority within the enterprise. A system for assigning trust credits to community members is considered future work.

After the feedback validation is performed, the ontology expert(s) **incorporate(s) its results into the enterprise-specific ontology**. Ontology experts do not interfere in feedback verification; they solely incorporate the final results into the ontology in a syntactically correct way, and link them with the core ontology. For the latter they use the contextual information that has been automatically attached to the ontology feedback, originating from the enterprise modelling cycle. Once a considerable amount of feedback is incorporated, a new enterprise-specific ontology version is proposed. The evolved ontology incorporates new concepts/relationships, updates lacking/incomplete and/or removes irrelevant ones, as the domain evolves or new business insights are reached. The evolved enterprise-specific ontology is subsequently communicated to the Ontology Storage and Suggestion Generation phase, where it will serve for (improved) suggestion generation to create (new) enterprise models.

3.2 Enterprise Modelling Cycle

The enterprise modelling cycle is instantiated every time an enterprise model is created for a specific project within the enterprise. Although all these enterprise models are developed in a different context and consequently have a different focus, the models should reuse the concepts, relations and axioms that are generally accepted within the firm and that are captured by the enterprise-specific ontology. This will make it easier to understand the developed models, will improve the traceability between the models and enhance their integration and interoperability. In the enterprise modelling cycle the enterprise engineer is therefore encouraged to use existing concepts by providing him with suggestions that are generated using the enterprise-specific ontology. In the first stage the modeller **selects the type of the enterprise model** to be created. Although our meta-method supports any kind of modelling language, it is important that the used modelling languages are analysed using the selected core ontology because the suggestion generation process relies hereupon.

During this **Enterprise Model Creation Phase** a modeller 1/ creates an enterprise model (e.g., a BPMN model), and 2/ provides feedback for the enterprise-specific ontology. Feedback may result e.g. from a missing concept/relationship, or a concept/relationship with a flaw (e.g., too restrictive or not restrictive enough). The phase starts with the modeller selecting the modelling construct that he wants to add to the enterprise model (e.g., a BPMN pool). The method responds by supplying the

modeller with suggestions of enterprise-specific ontology elements related to the selected construct. The modeller goes through the suggestions and provides feedback based on relevance for his particular purpose.

In case the modeller finds a relevant concept in the list, he thoroughly checks its definition and all other related information to make sure it precisely describes what he is looking for. If some information is found to be inaccurate, incorrect or missing, the modeller reports this. This feedback includes the concept itself, the error(s)/shortcoming(s) and a justification. If the concept is complete and correct, the modeller adds it to the enterprise model and the construct is automatically annotated with the relevant ontology concept. In case the modeller does not find any suitable concept in the suggestion list, he adds the modelling construct, naming it freely, without any relation with the enterprise-specific ontology. Subsequently, the modeller is asked to clarify the basic concept(s)/relationship(s) behind this modelling construct and a motivation, both of which are formulated as feedback about the ontology. When the modeller is finished with one construct, he continues the modelling process by selecting another construct. This iterative process continues until the model is finished.

The purpose of the **Enterprise Model Evaluation Phase** is to evaluate the resulting model quality, both quantitatively and qualitatively. Based on the evaluation results, the modeller may update his model, until it is judged to be of sufficient quality to be used in the enterprise. Feedback to the Ontology Engineering cycle is only done if the model attains a predefined quality, in order not to generate low quality feedback for the community members. If this is the case, feedback is generated and opened up for discussion by the community. Model quality is evaluated using a set of predefined quality measures that are selected based on model's usage and model type.

4 Conclusion

This paper introduces a reciprocal enterprise-specific ontology and enterprise modelling meta-method, which provides a unique symbiosis between enterprise modelling and ontology engineering. On one hand, it facilitates enterprise modellers to use an enterprise-specific ontology for developing enterprise models. While doing so, the modeller receives support in the form of suggestions based on the enterprise-specific ontology, and (automatically) annotates the developed models with shared, enterprise-specific ontology concepts. On the other hand, enterprise modellers give feedback on detected ontology flaws, i.e., incorrect, missing or incomplete ontology concepts or relationships. This feedback provides a vital input for participatory-based ontology engineering, in which ontologies evolve in consensus among the different (enterprise) stakeholders. In this article, we provided the reader with a general overview of the meta-method, its different phases, and the techniques used in every phase. Our method differs from existing research on enterprise ontology and enterprise models in the fact that it 1/ allows ontology improvement and evolution by means of feedback from enterprise models derived from real life needs, 2/ automatically annotates the developed enterprise models with enterprise-specific ontology concepts, 3/ assists the enterprise modelling process based on an enterprise-specific ontology.

In future research we discern two research avenues. First, we plan to investigate further instantiations of our meta-method. In particular, different target enterprise

modelling languages and their analysis with respect to different core ontologies will be investigated. Second, we plan to further develop the different phases of the meta-method. In particular, the suggestion generation algorithms will be further elaborated, and refined to cover different core ontologies on one hand, and different target enterprise modelling language on the other. We also foresee the development a platform where community members can negotiate upon feedback.

References

1. Bera, P., Burton-Jones, A., Wand, Y.: Guidelines for Designing Visual Ontologies to Support Knowledge Identification. *Mis Quarterly* 35, 883–908 (2011)
2. Geerts, G.L., McCarthy, W.E.: An Accounting Object Infrastructure for Knowledge Based Enterprise Models. *IEEE Intelligent Systems and Their Applications* 14, 89–94 (1999)
3. Gordijn, J., Akkermans, J.M.: E3-value: Design and Evaluation of e-Business Models. *IEEE Intelligent Systems* 16, 11–17 (2001)
4. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. *The Knowledge Engineering Review: Special Issue on Putting Ontologies to Use* 13, 31–89 (1998)
5. Hepp, M., Bachlechner, D., Siorpaes, K.: OntoWiki: Community-driven Ontology Engineering and Ontology Usage based on Wikis. In: Proceedings of the 2005 International Symposium on Wikis (WikiSym 2005), San Diego, California, USA (2005)
6. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: *Ontological Engineering*. Springer (2004)
7. Stirna, J., Persson, A., Sandkuhl, K.: Participative Enterprise Modeling: Experiences and Recommendations. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007)
8. Vernadat, F.: UEML: towards a unified enterprise modelling language. *International Journal of Production Research* 40, 4309–4321 (2002)
9. Lankhorst, M.: *Enterprise architecture at work: modelling, communication, and analysis*. Springer, Berlin (2005)
10. De Moor, A.: Ontology-Guided Meaning Negotiation in Communities of Practice Communication in Communities of Practice. In: Proc. of the Workshop on the Design for Large-Scale Digital Communities at the 2nd International Conference on Communities and Technologies, Milan, Italy (2005)
11. Aschoff, F.-R., Schmalhofer, F., van Elst, L.: Knowledge mediation: A procedure for the cooperative construction of domain ontologies. In: Motta, E., Shadbolt, N.R., Stutt, A., Gibbins, N. (eds.) EKAW 2004. LNCS (LNAI), vol. 3257, pp. 506–508. Springer, Heidelberg (2004)
12. Hepp, M., Siorpaes, K., Bachlechner, D.: Harvesting Wiki Consensus: Using Wikipedia Entries as Vocabulary for Knowledge Management. *IEEE Internet Computing* 11, 54–65 (2007)
13. Guarino, N.: Formal Ontology and Information Systems. In: International Conference on Formal ontology in Information Systems (FOIS 1998), pp. 3–15. IOS Press, Trento (1998)
14. Guizzardi, G., Wagner, G.: What's in a Relationship: An Ontological Analysis. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 83–97. Springer, Heidelberg (2008)
15. Gupta, U.G., Clarke, R.E.: Theory and applications of the Delphi technique: A bibliography (1975–1994). *Technological Forecasting and Social Change* 53, 185–211 (1996)

sonSQL: An Extensible Relational DBMS for Social Network Start-Ups^{*}

Zhifeng Bao, Jingbo Zhou, and Y.C. Tay

National University of Singapore

Abstract. There is now a proliferation of social network start-ups. This demonstration introduces **sonSQL**, a MySQL variant that aims to be the default off-the-shelf database system for managing their social network data.

1 Introduction

The mushrooming of online social networks is now as unstoppable as the spread of the Web. Their datasets are heterogeneous and, sometimes, huge. Whatever its size, such a network needs a database system to properly manage its data. However, a small start-up with limited database expertise may pick a non-extensible design that is hard to modify when more features are included as the social network grows. The database research community should help such start-ups by designing and engineering a robust and scalable system that is customized for managing social network data.

Our contribution to solving the problem is **sonSQL** (“son” for “social network”). We plan to develop sonSQL into the default database system for social networks. We start with the MySQL codebase, and restrict the conceptual schema to **sonSchema**, which is tailored for social networks [1].

A visitor to our demonstration will see their social network design automatically transformed into relational tables for users, products, interactions, etc. The transformation has a question-and-answer interface that does not require database expertise from the user. The visitor can use the interface to (1) construct a relational database schema for the social entities, products and interactions; (2) populate the tables with synthetic data for test queries; and (3) modify or update the schema.

2 Objectives

We now state our technical objectives, and briefly say how we aim to fulfill them. For clarity, we refer to the sonSQL user as an **SNcreator**, who uses sonSQL to create a social network **SN**; we refer to the users of SN as **SNusers**. An SN typically belongs to some **SNdomain** (entertainment, education, technology, etc.).

The first objective is (**O1**) *sonSQL should be based on a database system that is freely available, yet reasonably complete.* Most models of social networks use graphs,

* This research was supported in part by MOE Grant No. R-252-000-394-112 and carried out at the SeSaMe Centre, which is supported by the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO.

but we decided against using a graph database system. Our most compelling reason is this: A database system for social networks must have an expressive query language, query optimization, indices, integrity constraints, concurrency control, crash recovery, batch processing, etc. Implementing this list to bring a prototype to market is highly nontrivial for any database system, and the only ones to do so are relational DBMS.

We hence adopt a relational system for (O1), and start with MySQL as the codebase.

Our second objective is (**O2**) *sonSQL must be sufficiently general that it can cover most, if not all, current and foreseeable social networks*. Our strategy is to have a design that is *service-oriented*, i.e. the entities and relationships in the schema must correspond naturally to social network activity and services. We did an extensive survey of current social network services, and arrived at the following fundamental characterization [1]: (i) An SN records (explicitly or implicitly) *social network relationships*, such as Facebook friends, Sina Weibo followers and LinkedIn groups. (ii) SNcreators and SNusers can introduce *social products*, such as Cyworld games, Flickr photos and Renren blogs. (iii) SNusers have *social interactions*. These are dynamic and cumulative, and each is related to a *social product*, like tagging a photograph, accepting an invitation, etc. (iv) Products can have *relationships*, like *coupon* for a *sale*, *poll* for a *meeting*, etc.

This characterization focuses on social interaction, and explicitly points out the role of products. We then designed sonSchema in Fig. 1 to match the characterization. Here, we highlight some entities and relationships (details in [1]):

- **social_product_activity** links a user to a product via an activity (vote in a poll, buy a coupon, etc.).
- **private_msg** is a message that is visible only to the sender and receiver(s).
- **response2post** is a relationship between a tag and an image, a comment and another comment, etc. It is a special case of **social_product_relationship**.

sonSchema fulfills objective (O2) because it exhausts the list of entities (users and products) and relationships (user-user, user-product and product-product). It is a conceptual schema: **user** in Fig. 1 can be instantiated as a table for retailers and another for advertisers, while buying a coupon and registering for a course can be different instances of **social_product_activity**. For contrast, consider Drupal Gardens (<http://drupalgardens.com>), which is a software-as-a-service for designing and hosting websites. Its interface for constructing an SN has a fixed list of products (blog,

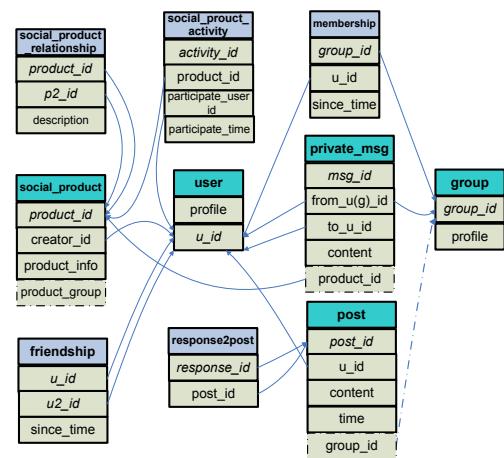


Fig. 1. The sonSchema conceptual schema. Table names are in **bold**, and edges point from foreign key to primary key. The 5 green tables are for entities, the 5 blue tables for relationships. Each table can have multiple logical instantiations.

forum, etc.) and services (follow, share, etc.) for SNcreator to choose from; she cannot customize these or create new ones, and the tables are pre-defined.

Many web services now offer social network applications. Such application data can be contained in a sonSchema instantiation that is separate from the legacy schema.

sonSchema's extensibility adds to our confidence in its generality (O2).

Our third objective is (**O3**) *a database novice should find it easy to use sonSQL to construct a schema for her social network design*. How can we provide an interface that requires little database expertise and minimal SNcreator effort, yet constructs a technically sound schema to match the SNcreator specification?

Fig. 2 shows the sonSQL architecture, with the interface at the top and MySQL at the bottom. The middle layer contains the *SN Constructor*, a *Module Tester* for the SNcreator to test the SN, and a *Data Generator* to populate the SN with test data.

Our solution to (O3) lies in form-based interaction. The forms are generated by a rule-based expert system, with help from a knowledge base. It has an inference engine called Entity Mapper that maps SNcreator's input into sonSchema entities and relationships. Its IC Verifier checks that the SNcreator's specifications satisfy sonSchema's integrity constraints. It then generates SQL DDL (or DML) scripts to construct (or update) MySQL tables.

Our fourth objective is (**O4**) *the schema should facilitate engineering for scalability*. sonSchema is in Boyce-Codd normal form, so its tables can be updated without requiring integrity checks that may be prohibitive in a distributed system under heavy workload. sonSchema is also hypergraph-acyclic, so it has a full reducer [2]. We are now studying the structure imposed by sonSchema on the space of all join trees, to identify bushy strategies for multi-way joins that execute faster than those produced by current optimizers [3]. We will also study the use of sonSchema's structure to design a concurrency control that provides strong consistency but without the ACID bottleneck.

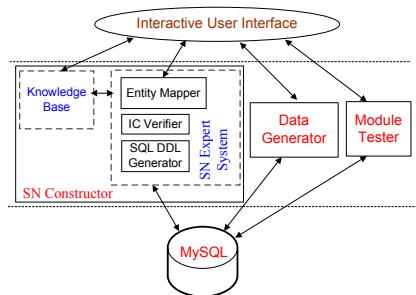


Fig. 2. sonSQL Architecture

3 The Demonstration

A visitor to our demonstration will be invited to assume the role of an SNcreator (see <http://sonsql.comp.nus.edu.sg/>).

Initial SN Characterization

sonSQL first presents a form like Fig. 3 for the SNcreator to browse a tree and identify the SNdomain (e.g. sports news or comic books) for her SN design. She can then specify the user categories (celebrities, authors, etc.) and social relationships (friendships, followers, etc.). With this information, sonSQL can create tables for **user**, **friendship**, **group** and **membership** in sonSchema.

Social Product Creation

Next, SNcreator sees a form for specifying products, including new ones (textbox in Fig. 3). When she selects a product (e.g. *coupon*), sonSQL checks its knowledge base and responds with options for *Producer* (advertiser? retailer?), *Consumer* (common SNuser? group?), possible related activities (*disseminate_coupon?*), and a high-level view of the SN (top-left in Fig. 4). Clicking on \otimes shows a mid-level pop-up view of relationships (top-right of Fig. 4). Clicking on an entity box shows a low-level view of the tables (bottom of Fig. 4). In this way, SNcreator iterates through the products in her SN design, thus specifying the details to sonSQL.

Modifications and Updates

Throughout, SNcreator can click on any part of the 3-level view to undo the latest change. After the changes are committed and even after the SN is deployed, SNcreator can call up similar forms to add new activities or remove some products, etc. sonSQL translates SNcreator's forms into SQL DDL or DML transactions, verifies that integrity constraints are satisfied, then sends the transactions to the MySQL backend for execution.

Testing the SN

sonSQL also has a form for the SNcreator to generate synthetic data, so she can run SQL queries to test her SN design.

References

1. Bao, Z., Tay, Y.C., Zhou, J.: sonSchema: A conceptual schema for social networks. In: Ng, W., Storey, V.C., Trujillo, J. (eds.) ER 2013. LNCS, vol. 8217, pp. 197–211. Springer, Heidelberg (2013)
2. Beeri, C., Fagin, R., Maier, D., Yannakakis, M.: On the desirability of acyclic database schemes. J. ACM 30(3), 479–513 (1983)
3. Huang, Q.: Optimizing PostgreSQL for social network databases. FYP report (December 2012)

Fig. 3. Form for initial SN characterization

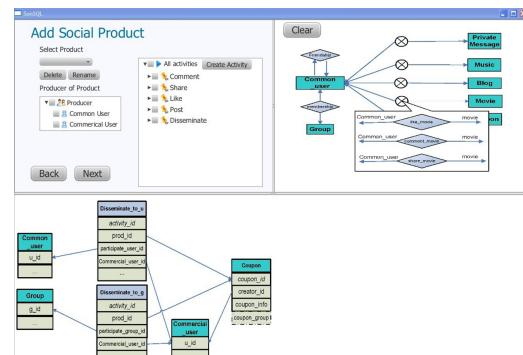


Fig. 4. 3-level views of the SN at the stage of adding social product

OntoDBench: Interactively Benchmarking Ontology Storage in a Database

Stéphane Jean¹, Ladjel Bellatreche¹, Carlos Ordonez²,
Géraud Fokou¹, and Mickaël Baron¹

¹ LIAS/ISAE-ENSMA, Futuroscope, France
`{jean, bellatreche, fokou, baron}@ensma.fr`

² University of Houston, Houston, U.S.A.
`ordonez@cs.uh.edu`

1 Introduction

Nowadays, all ingredients are available for developing domain ontologies. This is due to the presence of various types of methodologies for creating domain ontologies [3]. The adoption of ontologies by real life applications generates mountains of ontological data that need techniques and tools to facilitate their storage, management and querying. The database technology was one of these solutions. Several academic and industrial database management systems (DBMS) have been extended with features designed to manage and to query this new type of data (e.g., Oracle [10], IBM Sor [6] or OntoDB [1]). The obtained databases are called *semantic databases* ($\mathcal{S}\mathcal{D}\mathcal{B}$).

Five main characteristics differentiate $\mathcal{S}\mathcal{D}\mathcal{B}$ from traditional databases. **(i)** They store both ontologies and their instances in the same repository. **(ii)** Three main storage layouts are candidates for storing ontologies and their instances: vertical (triple table), horizontal (one table by class), binary (one table by property) [8], where each one has its own advantages and drawbacks. **(iii)** $\mathcal{S}\mathcal{D}\mathcal{B}$ have three main architectures. Systems such as Oracle [10] use the traditional databases architecture with two parts: *data schema part* and the *system catalog part*. In systems such as IBM Sor [6], the ontology is separated from its instances resulting in an architecture with three parts: the *ontology part*, the *data schema part* and the *system catalog part*. OntoDB [1] considers an architecture with *four parts*, where a new part called the *meta-schema part* is added as a system catalog for the ontology part. **(iv)** The ontology referencing $\mathcal{S}\mathcal{D}\mathcal{B}$ instances may be expressed in various formalisms (RDF, RDFS, OWL, etc.). **(v)** Ontology instances can be rather structured like relational data or be completely unstructured. Indeed, some concepts and properties of an ontology may not be used by a target application. As a consequence, if only a fragment of the domain ontology is used, ontology instances have a lot of *NULL* values. On the contrary, the whole ontology could be used resulting in relational-like ontology instances.

These characteristics make the development of $\mathcal{S}\mathcal{D}\mathcal{B}$ benchmarks challenging. Several benchmarks exist for $\mathcal{S}\mathcal{D}\mathcal{B}$ [7,9,5]. They present the following drawbacks: **(i)** they give contradictory results since they used datasets and queries with different characteristics. **(ii)** A gap exists between the generated datasets used by existing benchmarks and the real datasets as shown in [2]. **(iii)** The absence of an interactive tool to facilitate

the use of those benchmarks. **(iv)** The task of setting all benchmark parameters is *time consuming* for the DBA.

Recently, Duan et al. [2] introduced a benchmark generator to overcome the gap between the generated datasets of existing benchmarks and the real datasets. This benchmark generator takes has input the structuredness of the dataset to be generated. However, this approach has two limitations: **(i)** it is difficult to do experiments for the whole spectrum of structuredness. Thus the DBA has to do its own experiments generating a dataset with the desired structuredness, loading it in \mathcal{SDB} and executing queries and **(ii)** the generated dataset is not associated to queries that are similar to the real workload. Again the DBA has to define queries on the generated dataset which are similar to her/his real application (same selectivity factors, hierarchies, etc.). Instead of defining a dataset and workload conform to the target application, we propose an alternative benchmarking system called *OntoDBench* to evaluate \mathcal{SDB} . The main difference with previous benchmarking systems is that *OntoDBench* takes as input *the real datasets and workload* of the DBA instead of using a generated dataset and predefined set of queries. *OntoDBench* has two main functionalities. Firstly it evaluates the scalability of the real workload on the three main storage layouts of \mathcal{SDB} . Then, according to her/his functionality and scalability requirements, the DBA may choose the adequate \mathcal{SDB} . This functionality is based on a rewriting query module that translates input queries according to the different storage layouts and includes ontology reasoning. Secondly, *OntoDBench* offers the DBA the possibility to estimate and modify the characteristics of its input dataset (e.g., structuredness of ontology instances or size of the ontology hierarchy) and workload (e.g., number of joins or selectivity factors of selections). This functionality can be used by the DBA to check whether an existing benchmark (e.g., the DBpedia SPARQL benchmark [7]) uses a dataset and workload similar to those present in her/his application. It can also be used to predict the behavior of storage layouts if ontology data and/or queries change.

2 OntoDBench: Metrics and Demonstration Description

Metrics play a key role in benchmark systems. Usually they are used to generate data with particular characteristics. In *OntoDBench*, they are used both for modifying the real dataset and for checking if the results of an existing benchmark are relevant to the real dataset/workload that must be managed. Since \mathcal{SDBs} store both ontologies and their instances and execute semantic queries, three types of metrics must be considered.

Ontology metrics: they include metrics such as the number of classes, the number of properties by class or the size of class and property hierarchies. The ontology is also characterized by the fragment of Description Logics used (to characterize the complexity of reasoning).

Instance metrics: they include metrics such as the number of instances by class or the average number of properties associated with a subject. We also consider the structuredness of a dataset which is defined in [2] according to the number of NULL values in the dataset. This number can be computed with the following formula: $\#NULL = (\sum_c |P(C)| \times |I(C, D)| - Nt(D))$, where $|P(C)|$, $|I(C, D)|$ and $|Nt(D)|$ represent the

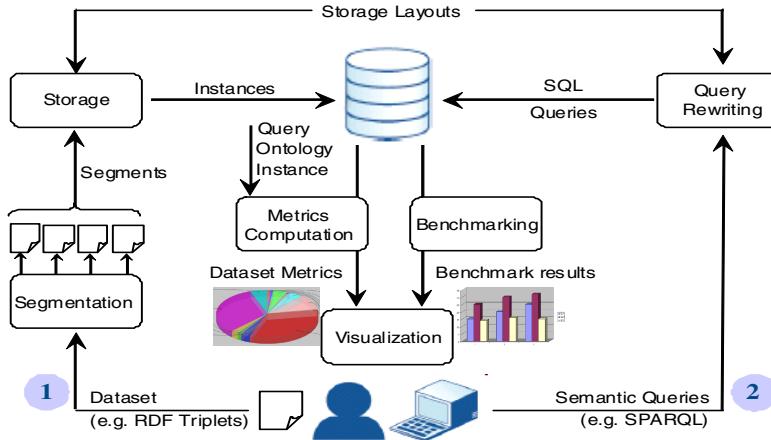


Fig. 1. The Components of our System

number of properties of the class C , the number of instances of C in the dataset D and the number of triples of the dataset D respectively.

This number of NULL can be computed to evaluate the structuredness of each class. This metric is called *coverage* of a class C in a dataset D , denoted $CV(C, D)$. It is defined as follows: $CV(C, D) = \frac{\sum_{p \in PC} OC(p, I(C, D))}{|P(C)| \times |I(C, D)|}$, where $OC(p, I(C, D))$ is the number of occurrences of a property p for the C instances of the dataset D .

A class can be more or less important in a dataset. If we denote τ the set of classes in the dataset, this weight WT is computed by: $WT(CV(C, D)) = \frac{|P(C)| + |I(C, D)|}{\sum_{C' \in \tau} (|P(C')| + |I(C', D)|)}$

This formula gives higher weights to the types with more instances and with a larger number of properties. The weight of a class combined with the coverage metric can be used to compute the structuredness of a dataset called *coherence*. The coherence of a dataset D composed of the classes τ (denoted $CH(\tau, D)$) is defined by: $CH(\tau, D) = \sum_{C \in \tau} WT(CV(C, D)) \times CV(C, D)$.

Query metrics: the considered semantic queries consist of conjunctive queries composed of selection and join operations. Thus these metrics include characteristics such as the selectivity factors of predicates or the number of join operations in the query.

DataSet Segmentation: loading a *big dataset* in the benchmark repository represents a real difficulty, especially when the size of these data exceeds the main memory. To overcome this problem, we add a new component allowing the DBA to segment the input files. We give the possibility to set the size of segments.

Dataset Storage: this module offers various possibilities to store the incoming dataset segments according to the three storage layouts. The loading process is executed with a multithreaded program (one thread for each segment). This process is achieved by (1) converting all the dataset in the N-Triples format since it maps directly to the vertical storage layout, (2) inserting each triple in the vertical storage layout and (3) loading the

dataset in the binary and horizontal storage layouts directly from the vertical storage layout (which was more efficient than reading again the input files). The conversion in the N-Triples format is done with the *Jena API*.

Metrics Computation: the second step of *OntoDBench* consists in computing the metrics of the dataset. This metric can be used to find the relevant benchmarks to the current scenario. Since the data are already in the database, the computation of most basic metrics is done with an SQL query. The computation of the coverage and coherence is more complex and is implemented with stored procedures. The metrics are automatically computed once the dataset is loaded and exported in a text file.

Query Rewriting Module: once the dataset is loaded in the database, the queries need to be translated according to the three storage layouts.

Benchmarking: with the previous query rewriting module, the workload under test can be executed on the three main storage layouts for ontology instances. The database buffers can have an influence on the query performance. Indeed, the first execution of a query is usually slower than the next executions due to the caching of data. As a consequence our benchmarking module takes as input the number of times the queries have to be executed.

Visualization: The benchmarking results (metrics, query processing, etc.) are stored in a text file. The DBA may visualize them via charts, histograms, etc. to facilitate their interpretation and exploitation for reporting and recommending a storage layout. The graphs are generated with the Java JFreeChart API¹.

Reasoning: Our system implements the two main reasoning approaches: (1) database saturation that consists in performing reasoning before query processing and to materialize all the deduced facts and (2) query reformulation that consists in performing reasoning during query processing by reformulating queries to include all virtual deduced facts. *OntoDBench* offers the DBA the possibility to test these two approaches. For the moment we have implemented the entailment rules of RDFS. For the database saturation approach, we use the PL/pgSQL database programming language to implement the 14 rules of RDFS. For the query reformulation approach, we have implemented the *reformulate algorithm* proposed in [4]. Figure 1 summarizes the different components of our system.

To validate our proposal we have done an implementation of *OntoDBench* (the source code is available at <http://www.lias-lab.fr/forge/projects/ontodbench/files>). We have used JAVA for the graphical user interface and PostgreSQL as a database storage. A demonstration video summarizing the different services offered by our benchmark is available at: <http://www.lias-lab.fr/forge/ontodbench/video.html>. The demonstration proposed in this paper consists in using *OntoDBench* on the LUBM dataset with a size ranging from 1K to 6 millions triplets and 14 queries. *OntoDBench* proposes a user friendly interface for the DBA so she/he can choose the size of segments, the storage layouts, etc. We demonstrate the following:

- the loading of huge amount of data by offering a *segmentation mechanism* that partition data in segments of a *fixed size*;

¹ www.jfree.org/jfreechart/

- the possibility to store the loaded data into the target DBMS according to the three main storage layouts (horizontal, vertical and binary);
- the computation of the dataset and workload metrics of the LUBM benchmark. The DBA may easily identify the degree (*high, medium and low*) of structuredness of the dataset. If the DBA is not satisfied with the obtained results, *OntoDBench* offers her/him the possibility to update the dataset to fit with her/his structuredness requirements;
- the query rewriting module of our system. We show the SQL translation of the LUBM queries on the three main storage layouts;
- the benchmarking of each query and the visualization of its result using graphs. The DBA may observe the query processing cost of all queries on the different storage layouts.

References

1. Dehainsala, H., Pierra, G., Bellatreche, L.: OntoDB: An Ontology-Based Database for Data Intensive Applications. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 497–508. Springer, Heidelberg (2007)
2. Duan, S., Kementsietsidis, A., Srinivas, K., Udrea, O.: Apples and oranges: a comparison of rdf benchmarks and real rdf datasets. In: SIGMOD, pp. 145–156 (2011)
3. Garcia-Alvarado, C., Chen, Z., Ordonez, C.: Ontocube: efficient ontology extraction using olap cubes. In: CIKM, pp. 2429–2432 (2011)
4. Goasdoué, F., Karanasos, K., Leblay, J., Manolescu, I.: View Selection in Semantic Web Databases. PVLDB Journal 5(2), 97–108 (2011)
5. Guo, Y., Pan, Z., Hefflin, J.: LUBM: A benchmark for OWL knowledge base systems. Journal of Web Semantics 3(2-3), 158–182 (2005)
6. Lu, J., Ma, L., Zhang, L., Brunner, J.-S., Wang, C., Pan, Y., Yu, Y.: Sor: a practical system for ontology storage, reasoning and search. In: VLDB, pp. 1402–1405 (2007)
7. Morsey, M., Lehmann, J., Auer, S., Ngonga Ngomo, A.-C.: DBpedia SPARQL Benchmark – Performance Assessment with Real Queries on Real Data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 454–469. Springer, Heidelberg (2011)
8. Ordonez, C., Cereghini, P.: Sqlem: Fast clustering in sql using the em algorithm. In: SIGMOD, pp. 559–570 (2000)
9. Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: SP2Bench: A SPARQL Performance Benchmark. In: ICDE, pp. 222–233 (2009)
10. Wu, Z., Eadon, G., Das, S., Chong, E.I., Kolovski, V., Annamalai, M., Srinivasan, J.: Implementing an inference engine for rdfs/owl constructs and user-defined rules in oracle. In: ICDE, pp. 1239–1248 (2008)

Specifying and Reasoning over Socio-Technical Security Requirements with STS-Tool

Elda Paja¹, Fabiano Dalpiaz², Mauro Poggianella¹,
Pierluigi Roberti¹, and Paolo Giorgini¹

¹ University of Trento, Italy

{elda.paja,mauro.poggianella,
pierluigi.roberti,paolo.giorgini}@unitn.it

² University of Toronto, Canada

dalpiaz@cs.toronto.edu

Abstract. We present the latest version of STS-Tool, the modelling and analysis support tool for STS-ml, an actor- and goal-oriented security requirements modelling language for socio-technical systems. STS-Tool allows designers to model a socio-technical system in terms of high-level primitives such as actor, goal, and delegation; to express security constraints over the interactions between the actors; and to derive security requirements once the modelling is done. The tool features a set of automated reasoning techniques for (i) checking if a given STS-ml model is well-formed, and (ii) determining if the specification of security requirements is consistent, that is, there are no conflicts among security requirements. These techniques have been implemented using disjunctive datalog programs. We have evaluated our tool through various industrial case studies.

1 Introduction

Today's systems are socio-technical, for they are an interplay of social actors (human and organisations) and technical components (software and hardware) that interact with one another for reaching their objectives and requirements [1]. Examples of these systems include healthcare systems, smart cities, critical infrastructure protection, next-generation of military protection and control, air traffic management control, etc.

The participants in a socio-technical system are autonomous, heterogeneous and weakly controllable. This raises up a number of security issues when they interact, especially when interaction involves the exchange of sensitive information: each participant would like to constrain, e.g., the way its information is to be manipulated by others, but has limited ways (due to uncontrollability) to do so.

When dealing with the security problem in socio-technical systems, it is not enough to consider technical mechanisms alone, because social aspects are a main concern. Considering the nature of the security problem in socio-technical systems, we have previously proposed STS-ml [2] (Socio-Technical Security modelling language), an actor- and goal-oriented security requirements modelling language for socio-technical systems, which relies on the idea of relating security requirements to interaction.

STS-ml allows stakeholders (reified as actors) to express *security needs* over interactions to constrain the way interaction is to take place, and uses the concept of *social commitment* [5] among actors to specify security requirements. For example, if a buyer

sends its personal data to a seller, the buyer may require the data not to be disclosed to third parties. In STS-ml, commitments are used to guarantee the satisfaction of *security needs*: one actor (*responsible*) commits to another (*requestor*) that it will comply with the required *security need*. In the previous example, the seller would commit not to disclose personal data to other parties.

We have previously shown [4] the use of *social commitments* in specifying security requirements. We have explained how STS-Tool¹, the graphical modelling and analysis support tool for STS-ml, supports the automated derivation of commitments.

Practical experiences with STS-Tool have shown [6] that, in large-scale scenarios, the security requirements posed by the various stakeholders are often inconsistent. Coping with such conflicts at requirements time avoids developing a non-compliant and hard-to-change system. In this work, our purpose is to illustrate the automated reasoning techniques (theoretically presented in [3]) that STS-Tool implements to detect inconsistencies among security requirements.

2 Demonstration Content

We illustrate STS-Tool by using an already modelled scenario from a case study on e-Government, developed as part of the EU FP7 project Aniketos. For a more interactive demo, we are going to show the tool in action by refining existing models (*Example 1*).

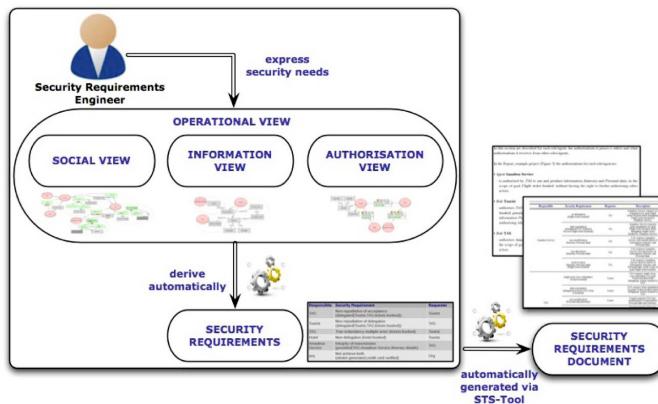


Fig. 1. From the operational view to security requirements

Example 1. Land selling involves finding a trustworthy buyer, and also exchanging several documents with various governmental bodies. The seller needs the municipality to certify that the land is in a residential zone. The land selling process is supported by an eGov application, through which the official contract (including the municipality's certification) is sent to the ministry (who has the right to object) and is archived.

Our demonstration covers the activities described in the following sub-sections.

¹ STS-Tool is freely available for download at <http://www.sts-tool.eu/>

2.1 Modelling with STS-Tool

We show how STS-Tool supports drawing STS-ml models of the system-to-be. STS-ml modelling is carried out by incrementally building three complementary views. As shown in Fig. 1, these views are: *social*, *information*, and *authorisation* view. The *security needs* constrain the interactions among actors. STS-Tool supports multi-view modelling by ensuring inter-view consistency by, for instance, propagating insertion or deletion of certain elements to all views, as well as ensuring diagram validity on the fly (well-formedness validity is checked while the models are being drawn). The tool also supports exporting the diagram (or the different views) to different image formats. We show how the tool supports the modelling process for the illustrating scenario.

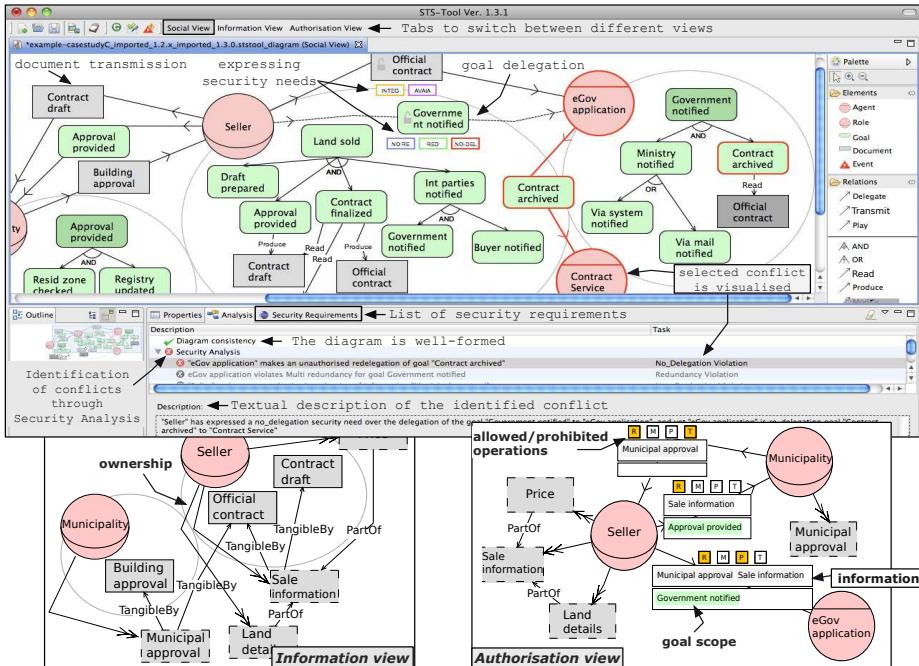


Fig. 2. Modelling, requirements derivation, and security analysis in STS-Tool

2.2 Specifying Security Requirements

We illustrate how security requirements are specified in terms of *social commitments*. Security requirements are automatically generated from a model as relationships between a *requester* and a *responsible* actor for the satisfaction of a *security need*. They can be sorted or filtered according to their different attributes. For instance, filtering the security requirements with respect to the *responsible* actor, highlights the actors that are responsible for satisfying the commitments (security requirements).

2.3 Reasoning about Security Requirements

We show the automated reasoning capabilities implemented in STS-Tool. The formal semantics of STS-ml [3] is defined in terms of possible actions and constraints on actions. STS-Tool supports the following checks: (i) well-formedness analysis to determine if the model complies with syntax restrictions (e.g., no cyclic decompositions), and (ii) security analysis, i.e., if there are potential conflicts of security requirements.

Well-formedness analysis is executed on demand, for its real-time execution would decrease the tool responsiveness. In Fig. 2, no well-formedness error was detected.

Security analysis is implemented in disjunctive Datalog and compares the possible actor behaviors that the model describes, against the security requirements that constrain possible behaviors. The results are enumerated in a table below the diagram (see Fig. 2). A textual description provides details on the identified conflicts.

2.4 Generating the Security Requirements Document

The modelling process terminates with *the generation of the security requirements document* (Fig. 1). This document is customisable: the analyst can choose among a number of model features to include in the report (e.g., including only a subset of the actors, concepts or relations). The diagrams are explained in detail providing textual and tabular descriptions of the models. The document is organised in sections, which the designer can decide to include or not in the document (see the website for an example).

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grants no. 257930 (Aniketos) and 256980 (NESSoS).

References

1. Dalpiaz, F., Giorgini, P., Mylopoulos, J.: Adaptive Socio-Technical Systems: a Requirements-driven Approach. *Requirements Engineering* 18(1), 1–24 (2013)
2. Dalpiaz, F., Paja, E., Giorgini, P.: Security requirements engineering via commitments. In: Proceedings of STAST 2011, pp. 1–8 (2011)
3. Paja, E., Dalpiaz, F., Giorgini, P.: Managing security requirements conflicts in socio-technical systems. In: Ng, W., Storey, V.C., Trujillo, J. (eds.) ER 2013. LNCS, vol. 8217, pp. 270–283. Springer, Heidelberg (2013)
4. Paja, E., Dalpiaz, F., Poggianella, M., Roberti, P., Giorgini, P.: STS-tool: Using commitments to specify socio-technical security requirements. In: Castano, S., Vassiliadis, P., Lakshmanan, L.V.S., Lee, M.L. (eds.) ER 2012 Workshops 2012. LNCS, vol. 7518, pp. 396–399. Springer, Heidelberg (2012)
5. Singh, M.P.: An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law* 7(1), 97–113 (1999)
6. Trösterer, S., Beck, E., Dalpiaz, F., Paja, E., Giorgini, P., Tscheligi, M.: Formative user-centered evaluation of security modeling: Results from a case study. *International Journal of Secure Software Engineering* 3(1), 1–19 (2012)

Lightweight Conceptual Modeling for Crowdsourcing

Roman Lukyanenko and Jeffrey Parsons

Faculty of Business Administration, Memorial University of Newfoundland
`{roman.lukyanenko, jeffreyp}@mun.ca`

Abstract. As more organizations rely on externally-produced information, an important issue is how to develop conceptual models for such data. Considering the limitations of traditional conceptual modeling, we propose a “lightweight” modeling alternative to traditional “class-based” conceptual modeling as typified by the E-R model. We demonstrate the approach using a real-world crowdsourcing project, NLNature.

Keywords: Conceptual Modeling, Information Quality, Ontology, Cognition.

1 Introduction

Organizations increasingly rely on externally produced information for decision making and operations support. This has led to the rise of *crowdsourcing* - wherein an organization engages external individuals to provide data for a specific purpose [1]. The use of crowdsourcing is expanding in diverse areas such as marketing, product development, collaborative mapping, crisis management, public policy, and scientific research.

A major data challenge in crowdsourcing is *conceptual modeling* [2]. Conceptual models commonly represent relevant knowledge about the application domain to guide information systems (IS) design, promote domain understanding and support communication [3]. Conceptual modeling is traditionally a user-driven activity. Users provide subject matter expertise and evaluate conceptual models. In contrast, in crowdsourcing, there are typically no constraints on who can contribute and engaging diverse audiences is highly desirable. As a result, some requirements and domain knowledge may originate from *system owners or sponsors*, but the actual data comes from distributed heterogeneous audiences. Many such users *lack domain expertise* (e.g., consumer product knowledge) and conceptualize the subject matter in ways incongruent with the views of project sponsors and other users. Unable to reach every potential contributor, modelers face extreme difficulty in trying to construct accurate and complete representations of modeled domains (see [2]).

2 NLNature: A Case of “Lightweight Conceptual Modeling”

Currently, no agreed-on conceptual modeling principles for crowdsourcing exist (for a discussion, see [2]). Several crowdsourcing projects implement and advocate prevailing

modeling approaches originally designed for modeling information requirements *within* organizations [4]. These approaches (e.g., E-R model), involve specifying *domain-specific abstractions* (e.g., classes, roles, actors applicable to a particular domain) that guide the design of IT artifacts (e.g., database schema, user interface, and programming code). Data creation and retrieval is then mediated by these artifacts (e.g., new entries must comply with the database schema). With the growth of distributed heterogeneous information (of which crowdsourcing is a type), several extensions to traditional conceptual modeling grammars have been proposed, but they continue to specify domain-specific constructs in advance of IS implementation [5]. We argue that, since crowdsourcing applications deal with inherently open and unpredictable phenomena, traditional modeling places unnecessary limitations on the kind of data users may provide, further constraining broader user engagement [2, 5].

We propose and evaluate the feasibility and relative advantages of a “lightweight modeling approach” to developing crowdsourcing IS. Under this approach, development proceeds by following an instance-based meta-model (see Fig. 1) informed by fundamental ontological and cognitive principles assumed to be broadly applicable across domains [7]. Data collection and retrieval are then powered by the instance-based data model [7]. In practice this means that users no longer need to classify instances of interest (e.g., birds, galaxies, material assets) and instead provide attributes of the observed instances. Different users can supply different attributes for the same instance. Failure to agree on classes or even attributes is no longer problematic as both convergence and divergence of views is accommodated: any relevant attribute can be captured. The attributes can be queried to infer classes of interest. Classes and other domain-specific abstractions are not necessary before implementing such an IS.

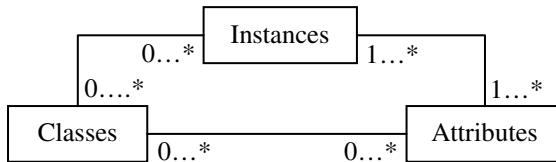


Fig. 1. Instance-based meta-model

During information requirements analysis, modelers can elicit attributes for a sample of instances in a domain from a sample of potential users. Analyzing these attributes can suggest data requirements and, for example, uncover the extent of convergence and divergence of user views. Analysis of user attributes can inform design choices (e.g., whether to constrain attributes to an authoritative list or leave it open), help developers to better understand the domain, and support communication during development.

To provide proof of concept and empirically evaluate the “lightweight modeling” approach, we developed an instance-based version of an existing crowdsourcing project, NLNature (www.nlnature.com). The project maps biodiversity of a region in North America (territory of over 150,000 square miles) using sightings of plants and animals provided by ordinary people. This data is then made available for researchers

in various disciplines (e.g., biology). NLNature started in 2009 and originally employed abstraction-driven modeling. The authors of this paper in tandem with the biologists (i.e., project sponsors) determined that quality and level of participation were below expectations and identified “heavyweight” (class-based) modeling as a root cause. As an alternative, the authors developed a set of modeling principles for crowdsourcing and implemented them in NLNature. In particular, we adopted the instance-based meta-model (in Fig. 1), collected a sample of attributes from potential citizen scientists, employed the instance-based data model for collection and storage of data, and developed a flexible user interface.

The instance-based version of NLNature permits citizen scientists to describe their sightings using attributes without having to classify observed instances (see, e.g., Fig. 2). This contrasts with the prevailing practice in citizen science to employ traditional domain-specific modeling [4] and ask contributors to classify (positively identify) observed instances according to biological species (e.g., American robin). We argue the instance-based version of NLNature should result in high quality of user-generated content and promote user participation.¹

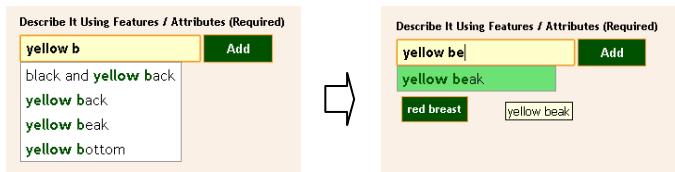


Fig. 2. Example of an attribute-based data collection on NLNature

3 Demonstration Highlights

During the demonstration session, we will provide the motivation for “lightweight conceptual modeling” paradigm and suggest some of the ways this approach can be implemented. The demonstration will focus on the following themes:

- We will discuss each phase of development under the “lightweight modeling” approach and provide illustrations of relevant artifacts (e.g., screenshots of instance-based data model objects, user interface, data extracts, and use live NLNature for an interactive demonstration)
- As “lightweight modeling” skips a major part of information requirements analysis, we will discuss what this means for analysis and what steps we took during this phase
- We will discuss the instance-based data model and explore ontological and cognitive considerations in choosing data models for “lightweight modeling”
- We will discuss the challenges “lightweight modeling” creates for interface design and use NLNature as an example of how interfaces can be developed.

¹ We are conducting empirical evaluation of the “lightweight conceptual modeling” approach by examining its impact on information quality and level of user participation. The discussion of the empirical evaluation is outside the scope of this demonstration and will be reported elsewhere.

4 Conclusion

The demonstration presents an opportunity to better understand fundamental issues in crowdsourcing, conceptual modeling, information quality, and system use. Using NLNature, we demonstrate application of “lightweight conceptual modeling” in an authentic setting. While the project is focused on crowdsourcing, we believe the ideas generalize to other heterogeneous environments (e.g., social media, flexible organizational settings, flexible inter-organizational systems, semantic web).

References

1. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing Systems on the World-Wide Web. *Communications of the ACM* 54, 86–96 (2011)
2. Lukyanenko, R., Parsons, J.: Conceptual Modeling Principles for Crowdsourcing. In: *Proceedings of the 1st International Workshop on Multimodal Crowd Sensing*, pp. 3–6 (2012)
3. Wand, Y., Weber, R.: Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research* 13, 363–376 (2002)
4. Wiggins, A., Bonney, R., Graham, E., Henderson, S., Kelling, S., LeBuhn, G., Litauer, R., Lots, K., Michener, W., Newman, G.: Data management guide for public participation in scientific research. *DataOne Working Group*, 1–41 (2013)
5. Lukyanenko, R., Parsons, J.: Is Traditional Conceptual Modeling Becoming Obsolete? In: *12th Symposium on Research in Systems Analysis and Design*, pp. 1–6 (2013)
6. Mylopoulos, J.: Information Modeling in the Time of the Revolution. *Information Systems* 23, 127–155 (1998)
7. Parsons, J., Wand, Y.: Emancipating Instances from the Tyranny of Classes in Information Modeling. *ACM Transactions on Database Systems* 25, 228–268 (2000)

Author Index

- Aberer, Karl 212
Adams, Michael 174
Alkhaldi, Nadejda 487
Almeida, João Paulo A. 327
Aufaure, Marie-Aude 9
Balke, Wolf-Tilo 298
Bao, Zhifeng 197, 241, 495
Barendsen, Erik 74
Baron, Mickaël 499
Bellatreche, Ladjel 499
Branco, Moisés C. 130
Casteleyn, Sven 487
Chin Seng, Teo 438
Cruzes, Daniela S. 414
Dalpiaz, Fabiano 33, 270, 504
de Castro, María Valeria 429
El Maarry, Kinda 298
Embley, David W. 1
Falbo, Ricardo de Almeida 327
Ferrarotti, Flavio 227
Fileto, Renato 342
Fillottrani, Pablo Rubén 313
Fokou, Géraud 499
Francesconi, Fabiano 33
Franch, Xavier 463
Gailly, Frederik 487
Gal, Avigdor 130, 212
Gilbert, Lester 161
Giorgini, Paolo 270, 504
Gmünder, Tobias 284
Gonzalez-Perez, Cesar 96, 479
Guizzardi, Giancarlo 327, 463, 471
Guizzardi, Renata S.S. 327, 463
Harzmann, Josefine 121
Henderson-Sellers, Brian 96, 479
Hengeveld, Sytse 74
Herzberg, Nico 146
Hoppenbrouwers, Stijn 74
Hoque, Zahirul 174
Ingolfo, Silvia 47
Jean, Stéphane 499
Kaschek, Roland 88
Keet, C. Maria 313
Khan, Shakil M. 19
Khovalko, Oleh 146
Knuplesch, David 106
Krüger, Marcelo 342
Kumar, Akhil 106
Le, Thuy Ngoc 356
Le, Van Bao Tran 227
Lee, Dik Lun 372
Lee, Mong Li 241
Leone, Stefania 284
Leung, Kenneth Wai-Ting 372
Levy, Eliezer 212
Li, Luochen 356
Liaskos, Sotirios 19
Liddle, Stephen W. 1
Ling, Tok Wang 241, 356
Link, Sebastian 227
Liu, Yuchen 372
Lofi, Christoph 298
Loucopoulos, Pericles 446
Low, Graham 479
Lu, Jiaheng 356
Lukyanenko, Roman 61, 508
Ly, Linh Thao 106
Manousis, Petros 182
Marcos, Esperanza 429
Martínez Ferrandis, Ana M^a 471
Mayr, Heinrich C. 403
McBride, Tom 479
Mecca, Giansalvatore 255
Meyer, Andreas 121, 146
Michael, Judith 403
Miklós, Zoltán 212
Mylopoulos, John 19, 33, 47

- Natvig, Marit K. 414
Norrie, Moira C. 284
- Olivé, Antoni 395
Ordonez, Carlos 499
Ouyang, Chun 174
- Paja, Elda 270, 504
Papastefanatos, George 182
Parsons, Jeffrey 61, 508
Partridge, Chris 96
Pastor, Joan Antoni 395
Pastor López, Oscar 471
Pelekis, Nikos 342
Perini, Anna 47
Poels, Geert 454
Poggianella, Mauro 504
Purao, Sandeep 438
- Quoc Viet Nguyen, Hung 212
- Reichert, Manfred 106
Reijers, Hajo A. 174
Renso, Chiara 342
Rinderle-Ma, Stefanie 106
Roberti, Pierluigi 504
Roelens, Ben 454
Rosemann, Michael 174
Rull, Guillem 255
- Santiago, Iván 429
Santoro, Donatello 255
Schultz, Martin 138
Shafran, Victor 212
- Sheetrit, Eitam 130
Siena, Alberto 47
Soutchanski, Mikhail 19
Sun, Jie 446
Susi, Angelo 47
- Tangworakitthaworn, Preecha 161
Tay, Y.C. 197, 495
Teniente, Ernest 255
ter Hofstede, Arthur H.M. 174
Theodoridis, Yannis 342
Tort, Albert 395
- van der Aalst, Wil M.P. 174
Vara, Juan Manuel 429
Vassiliadis, Panos 182
Vennesland, Audun 414
- Wagner, Gerd 327
Weidlich, Matthias 130, 212
Werner, Michael 387
Weske, Mathias 121, 146
Wieringa, Roel 463
Wijaya, Tri Kurniawan 212
Wills, Gary B. 161
Wilmont, Ilona 74
Wu, Alfred 438
Wu, Huayu 356
Wynn, Moe T. 174
- Zäschke, Tilmann 284
Zeng, Zhong 241
Zhao, Liping 446
Zhou, Jingbo 197, 495