LNCS 7046

Robert Meersman
Tharam Dillon
Pilar Herrero (Eds.)

On the Move to Meaningful Internet Systems: OTM 2011 Workshops

Confederated International Workshops and Posters: EI2N+NSF ICE, ICSP+INBAST, ISDE, ORM, OTMA, SWWS+MONET+SeDeS, and VADER 2011 Hersonissos, Crete, Greece, October 2011, Proceedings

Springer

Lecture Notes in Computer Science 7046

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

Robert Meersman   Tharam Dillon
Pilar Herrero (Eds.)

On the Move to Meaningful Internet Systems: OTM 2011 Workshops

Confederated International Workshops and Posters:
EI2N+NSF ICE, ICSP+INBAST, ISDE, ORM,
OTMA, SWWS+MONET+SeDeS, and VADER 2011
Hersonissos, Crete, Greece, October 17-21, 2011
Proceedings

Springer

Volume Editors

Robert Meersman
Vrije Universiteit Brussel (VUB), STAR Lab
Bldg G/10, Pleinlaan 2, 1050 Brussel, Belgium
E-mail: meersman@vub.ac.be

Tharam Dillon
Curtin University of Technology, DEBII
Technology Park, De Laeter Way, Bentley, WA 6102, Australia
E-mail: t.dillon@curtin.edu.au

Pilar Herrero
Universidad Politécnica de Madrid, Facultad de Informática
Campus de Montegancedo S/N, 28660 Boadilla del Monte, Madrid, Spain
E-mail: pherrero@fi.upm.es

ISSN 0302-9743
e-ISSN 1611-3349
ISBN 978-3-642-25125-2
e-ISBN 978-3-642-25126-9
DOI 10.1007/978-3-642-25126-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011940131

CR Subject Classification (1998): H.4, C.2, H.3, I.2, D.2, H.5

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

© Springer-Verlag Berlin Heidelberg 2011
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, 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.

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)

Volume Editors

Robert Meersman
Tharam Dillon
Pilar Herrero

EI2N + NSF ICE 2011

Hervé Panetto
Qing Li
J. Cecil

ICSP 2011

Hervé Panetto
Richard Mark Soley

INBAST 2011

Thomas Moser
Yan Tang

ISDE 2011

Alok Mishra
Jürgen Münch
Ricardo Colomo Palacios
Deepti Mishra

MONET 2011

Patrizia Grifoni
Fernando Ferri
Irina Kondratova
Arianna D'Ulizia

ORM 2011

Terry Halpin
Herman Balsters

OTMA 2011

Peter Spyns
Anja Metzner

SeDeS 2011

Yan Tang
Jan Vanthienen

SWWS 2011

Ernesto Damiani
Elizabeth Chang
Paolo Ceravolo
Omar Khadeer Hussain

VADER 2011

Miguel Angel Pérez-Toledano
Carlos E. Cuesta
Renaud Pawlak
Javier Cámara

General Co-chairs' Message for OnTheMove 2011

The OnTheMove 2011 event in Heraklion, Crete, held during October 17-21, further consolidated the growth of the conference series that was started in Irvine, California, in 2002, and held in Catania, Sicily, in 2003, in Cyprus in 2004 and 2005, in Montpellier in 2006, in Vilamoura in 2007 and 2009, in Monterrey, Mexico, in 2008, and in Heraklion 2010. The event continues to attract a diversified and representative selection of today's worldwide research on the scientific concepts underlying new computing paradigms, which, of necessity, must be distributed, heterogeneous, and autonomous yet meaningfully collaborative. Indeed, as such large, complex, and networked intelligent information systems become the focus and norm for computing, there continues to be an acute and even increasing need to address and discuss face to face in an integrated forum the implied software, system, and enterprise issues as well as methodological, semantic, theoretical, and application issues. As we all realize, email, the Internet, and even video conferences are not by themselves sufficient for effective and efficient scientific exchange.

The OnTheMove (OTM) Federated Conference series has been created to cover the scientific exchange needs of the community/ies that work in the broad yet closely connected fundamental technological spectrum of Web-based distributed computing. The OTM program every year covers data and Web semantics, distributed objects, Web services, databases, information systems, enterprise workflow and collaboration, ubiquity, interoperability, mobility, grid and high-performance computing.

OnTheMove does not consider itself a so-called multi-conference event but instead is proud to give meaning to the "federated" aspect in its full title : it aspires to be a primary scientific meeting place where all aspects of research and development of Internet- and intranet-based systems in organizations and for e-business are discussed in a scientifically motivated way, in a forum of (loosely) interconnected workshops and conferences. This tenth edition of the OTM Federated Conferences event therefore once more provided an opportunity for researchers and practitioners to understand and publish these developments within their individual as well as within their broader contexts. To further promote synergy and coherence, the main conferences of OTM 2011 were conceived against a background of three interlocking global themes:

- Virtual ("Cloud") Computing Infrastructures and Security
- The Internet of Things, and Semantic Web 2.0
- Collaborative ("Social") Computing for the Enterprise

Originally the federative structure of OTM was formed by the co-location of three related, complementary, and successful main conference series: DOA

(Distributed Objects and Applications, since 1999), covering the relevant infrastructure-enabling technologies, ODBASE (Ontologies, DataBases and Applications of Semantics, since 2002) covering Web semantics, XML databases and ontologies, and CoopIS (Cooperative Information Systems, since 1993) covering the application of these technologies in an enterprise context through, e.g., workflow systems and knowledge management. In 2007 the IS workshop (Information Security) was added to try cover also the specific issues of security in complex Internet-based information systems, and in this 2011 edition these security aspects became merged with DOA under the umbrella description of "secure virtual infrastructures," or DOA-SVI. Each of the main conferences specifically seeks high-quality contributions and encourages researchers to treat their respective topics within a framework that incorporates jointly (a) theory, (b) conceptual design and development, and (c) applications, in particular case studies and industrial solutions.

Following and expanding the model created in 2003, we again solicited and selected quality workshop proposals to complement the more "archival" nature of the main conferences with research results in a number of selected and more "avant-garde" areas related to the general topic of Web-based distributed computing. For instance, the so-called Semantic Web has given rise to several novel research areas combining linguistics, information systems technology, and artificial intelligence, such as the modeling of (legal) regulatory systems and the ubiquitous nature of their usage. We were glad to see that six of our earlier successful workshops (EI2N, SWWS, ORM, MONET, ISDE, SeDeS) re-appeared in 2011 with in some cases a fourth or even fifth edition, often in alliance with other older or newly emerging workshops, and that three brand-new independent workshops could be selected from proposals and hosted: INBAST, RASEP, and VADER. (INBAST was merged with the new Industry Track, under the auspicious leadership of Hervé Panetto and OMG's Richard Mark Soley.)

We are also proud in particular to note the co-sponsorship of the US National Science Foundation (NSF) for the EI2N workshop (also initiated by Hervé), and which increasingly profiles itself as a successful incubator for new "CoopIS-related" research aspects and topics. Our OTM registration format ("one workshop buys all") actively intends to stimulate workshop audiences to productively mingle with each other and, optionally, with those of the main conferences.

We were again most happy to see once more in 2011 the number of quality submissions for the OnTheMove Academy (OTMA, formerly called Doctoral Consortium Workshop), our "vision for the future" in research in the areas covered by OTM, managed by a dedicated team of collaborators led by Peter Spyns and Anja Schanzenberger, and of course by the OTMA Dean, Erich Neuhold, responsible for our unique interactive formula to bring PhD students together. In the OTM Academy, PhD research proposals are submitted for peer review; selected submissions and their approaches are (eventually) presented by the students in front of a wider audience at the conference, and independently and extensively analyzed and discussed in front of the audience by a panel of senior professors.

As said, all three main conferences and the associated workshops shared the distributed aspects of modern computing systems, and the resulting application pull created by the Internet and the so-called Semantic Web. For DOA-SVI 2011, the primary emphasis stayed on the distributed object infrastructure and its virtual and security aspects; for ODBASE 2011, the focus became the knowledge bases and methods required for enabling the use of formal semantics in Web-based databases and information systems; for CoopIS 2011, the focus as usual was on the interaction of such technologies and methods with management issues, such as occur in networked organizations and enterprises. These subject areas overlap in a scientifically natural fashion and many submissions in fact also treated an envisaged mutual impact among them. As with the earlier editions, the organizers wanted to stimulate this cross-pollination by a "shared" program of famous keynote speakers around the chosen themes. We were quite proud to announce:

– Amit Sheth, Wright State University, Ohio, USA
– Schahram Dustdar, Vienna University of Technology, Austria
– Siani Pearson, Hewlett-Packard Laboratories, Bristol, UK
– Niky Riga, Raytheon BBN Technologies, Massachusetts, USA

We received a total of 141 submissions for the three main conferences and 104 submissions in total for the workshops. The numbers are comparable with those for 2010. Not only may we indeed again claim success in attracting an increasingly representative volume of scientific papers, many from the USA and Asia, but these numbers of course allow the Program Committees to compose a high-quality cross-section of current research in the areas covered by OTM. In fact, the Program Chairs of the CoopIS 2011 conferences decided to accept only approximately 1 paper for each 5 submissions, while the ODBASE 2011 PC accepted about the same number of papers for presentation and publication as in 2009 and 2010 (i.e., average 1 paper out of 3-4 submitted, not counting posters). For the workshops and DOA-SVI 2011 the acceptance rate varies but the aim was to stay consistently at about 1 accepted paper for 2-3 submitted, and this of course subordinated to peer assessment of scientific quality.

As usual we have separated the proceedings into three volumes with their own titles, two for the main conferences and one for the workshops, and we are again most grateful to the Springer LNCS team in Heidelberg for their professional suggestions and meticulous collaboration in producing the files for downloading on the USB sticks.

The reviewing process by the respective Program Committees was again performed very professionally, and each paper in the main conferences was reviewed by at least three referees, with arbitrated email discussions in the case of strongly diverging evaluations. It may be worth emphasizing that it is an explicit OnTheMove policy that all conference Program Committees and Chairs make their selections completely autonomously from the OTM organization itself. Like last year, paper proceedings were on separate request and order this year, and incurred an extra charge.

The General Chairs are once more especially grateful to the many people directly or indirectly involved in the set-up of these federated conferences. Not everyone realizes the large number of persons that need to be involved, and the huge amount of work, commitment, and in the uncertain economic and funding climate of 2011 certainly also financial risk, the organization of an event like OTM entails. Apart from the persons in their roles mentioned above, we therefore wish to thank in particular our eight main conference PC Co-chairs:

- CoopIS 2011: Manfred Reichert, Akhil Kumar, Qing Li
- ODBASE 2011: Manfred Hauswirth, Pascal Hitzler, Mukesh Mohania
- DOA-SVI 2011: Ernesto Damiani, Doug Schmidt, Beng Chin Ooi

And similarly the 2011 OTMA and Workshops PC (Co-)chairs (in arbitrary order): Hervé Panetto, Qing Li, J. Cecil, Thomas Moser, Yan Tang (2x), Alok Mishra, Jürgen Münch, Ricardo Colomo Palacios, Deepti Mishra, Patrizia Grifoni, Fernando Ferri, Irina Kondratova, Arianna D'Ulizia, Terry Halpin, Herman Balsters, Almudena Alcaide, Naoki Masuda, Esther Palomar, Arturo Ribagorda, Yan Zhang, Jan Vanthienen, Ernesto Damiani (again), Elizabeth Chang, Paolo Ceravolo, Omar Khadeer Hussain, Miguel Angel Pérez-Toledano, Carlos E. Cuesta, Renaud Pawlak, Javier Cámara, Stefanos Gritzalis, Peter Spyns, Anja Metzner, Erich J. Neuhold, Alfred Holl, and Maria Esther Vidal.

All of them together with their many PC members, performed a superb and professional job in managing the difficult yet existential process of peer review and selection of the best papers from the harvest of submissions. We are all also grateful to our supremely competent and experienced Conference Secretariat and technical support staff in Antwerp and Guadalajara, Jan Demey and Daniel Meersman, and last but certainly not least to our proceedings production team in Perth (DEBII-Curtin University) this year led by Christopher Jones.

The General Co-chairs acknowledge with gratitude the academic freedom, logistic support, and facilities they enjoy from their respective institutions, Vrije Universiteit Brussel (VUB), Curtin University, Perth, Australia, and Universidad Politécnica de Madrid (UPM), without which such an enterprise quite simply would not be feasible. We do hope that the results of this federated scientific enterprise contribute to your research and your place in the scientific network... We look forward to seeing you again at next year's event!

August 2011

Robert Meersman
Tharam Dillon
Pilar Herrero

Organization

OTM (On The Move) is a federated event involving a series of major international conferences and workshops. These proceedings contain the papers presented at the OTM 2011 Federated workshops, consisting of three conferences, namely, CoopIS 2011 (Cooperative Information Systems), DOA-SVI 2011 (Secure Virtual Infrastructures), and ODBASE 2011 (Ontologies, Databases and Applications of Semantics).

Executive Committee

General Co-chairs

Robert Meersman	VU Brussels, Belgium
Tharam Dillon	Curtin University of Technology, Australia
Pilar Herrero	Universidad Politécnica de Madrid, Spain

OnTheMove Academy Dean

Erich Neuhold	University of Vienna, Austria

EI2N+NSF ICE 2011 PC Co-chairs

Hervé Panetto	Nancy University, France
Qing Li	Tsinghua University, P. R. China
J. Cecil	Oklahoma State University, USA

ICSP 2011 PC Co-chairs

Hervé Panetto	Nancy University, France
Richard Mark Soley	OMG, USA

INBAST 2011 PC Co-chairs

Thomas Moser	Vienna University of Technology, Austria
Yan Tang	Vrije Universiteit Brussel, Belgium

ISDE 2011 PC Co-chairs

Alok Mishra	Atilim University, Turkey
Jürgen Münch	Fraunhofer (IESE), Kaiserslautern, Germany
Ricardo Colomo Palacios	Universidad Carlos III de Madrid, Spain
Deepti Mishra	Atilim University, Turkey

ORM 2011 PC Co-chairs

Terry Halpin	LogicBlox, Australia and INTI International University, Malaysia
Herman Balsters	University of Groningen, The Netherlands

SWWS 2011 PC Co-chairs

Ernesto Damiani	University of Milan, Italy
Elizabeth Chang	DEBII, Curtin University, Australia
Paolo Ceravolo	University of Milan, Italy
Omar Khadeer Hussain	DEBII, Curtin University, Australia

MONET 2011 PC Co-chairs

Patrizia Grifoni	IRPPS, Rome, Italy
Fernando Ferri	IRPPS, Rome, Italy
Irina Kondratova	NRC Institute for Information Technology, Fredericton, Canada
Arianna D'Ulizia	IRPPS, Rome, Italy

SeDeS 2011 PC Co-chairs

Yan Tang	STARLab, Vrije Universiteit Brussel, Belgium
Jan Vanthienen	LIRIS, Katholieke Universiteit Leuven, Belgium

VADER 2011 PC Co-chairs

Miguel Angel Pérez-Toledano	University of Extremadura, Spain
Carlos E. Cuesta	Rey Juan Carlos University, Spain
Renaud Pawlak	ISEP, France
Javier Cámara	University of Coimbra, Portugal

Publication Chair

Christopher Jones	Curtin University of Technology, Australia

Publicity-Sponsorship Chair

Ana-Cecilia Martinez Barbosa	DOA Institute, Belgium

Logistics Team

Daniel Meersman	Head of Operations
Ana-Cecilia Martinez Barbosa	
Jan Demey	

EI2N+NSF ICE 2011 Program Committee

Afsarmanesh
Giuseppe Berio
Peter Bernus
Xavier Boucher
Nacer Boudjlida
Luis Camarinha-Matos
Cecil Joe
Vincent Chapurlat
Yannis Charalabidis
David Chen
Adrian Curaj
Michele Dassisti
Andres Garcia Higuera
François Kiefer
Ricardo Goncalves
Ted Goranson
Uma Jayaram
Charlotta Johnsson
John Krogstie
Qing Li
Peter Loos
Juan-Carlos Mendez
Pierre-Alain Millet
Arturo Molina
Yannick Naudet
Ovidiu Noran
Angel Ortiz
Hervé Panetto
Jin woo Park
Michaël Petit
Wolfgang Prinz
David Romero
Camille Salinesi
Czeslaw Smutnicki
Aurelian Mihai Stanescu
Kamelia Stefanova
Mike Sobolewski
Richard Soley
Janusz Szpytko
Zahir Tari
Miroslav Trajanovic
Bruno Vallespir
François B. Vernadat
Georg Weichhart
Lawrence Whitman
Milan Zdravkovic

ICSP 2011 Program Committee

Sinuhe Arroyo
Peter Benson
M. Brian Blake
Emmanuel Blanvilain
Serge Boverie
Dennis Brandl
Christoph Bussler
Luis Camarinha-Matos
Jorge Cardoso
Vincent Chapurlat
Yanni Charalabidis
Stijn Christiaens
Francesco Danza
Michele Dassisti
Piero De Sabbata
Marc Delbaere
Marin Dimitrov
Dominique Ernadote
Claude Feliot
Donald Ferguson
Stephen Flinter
George Fodor
Andres Garcia Higuera
Pascal Gendre
Ricardo Goncalves
Ted Goranson
Tony Hutinet
Cecil Joe
Florian Kerschbaum
Laurent Liscia
Peter Loos
Richard Martin

Juan-Carlos Mendez
Arturo Molina
Yannick Naudet
Yasuyuki Nishioka
Hervé Panetto
Jin woo Park
Andrea Persidis
Roger Roberts
Pascal Roques
Ayelet Sapir
Richard Soley
Axel Uhl
François B. Vernadat
Steve Vinoski
Alex Wahler
Lawrence Whitman
Milan Zdravkovic
Martin Zelm
Detlef Zhlke

INBAST 2011 Program Committee

Abraham Bernstein
Olawande Daramola
Sergio de Cesare
Pieter De Leenheer
Deepak Dhungana
Christian Frühwirth
Fausto Giunchiglia
Mustafa Jarrar
Olga Kovalenko
Richard Mordinyi
Petr Novák
Inah Omoronyia
Szabolcs Rosznyai
Peter Spyns
Wikan Sunindyo
Ludger van Elst
Manuel Wimmer
Dietmar Winkler
Zhenzhen Zhao

ISDE 2011 Program Committee

Silvia Abrahao
Ian Allison
M. Ali Babar
Barbara Carminati
Cagatay Catal
Kai Chen
Juan Garbajosa
Cigdem Gencel
Amar Gupta
Orit Hazzan
Jukka Kääriäinen
Allen E. Milewski
Nils. B. Moe
Mahmood Niazi
Nilay Oza
Francisco J.G. Peñalvo
Srini Ramaswamy
Ita Richardson
Kassem Saleh
Pedro Soto-Acosta
Pierre F. Tiako
Anil Kumar Tripathi
João Varajão
June Verner
Deo Prakash Vidyarthi
Charles Wallace
Bernard Wong
Qing Yao
Liguo Yu

ORM 2011 Program Committee

Herman Balsters
Scott Becker
Linda Bird
Anthony Bloesch
Peter Bollen
Andy Carver
Matthew Curland
Dave Cuyler
Necito Dela Cruz
Gordon Everest
Ken Evans
Pat Hallock
Terry Halpin
Clifford Heath
Stijn Hoppenbrouwers
Mike Jackson
Mustafa Jarrar
Tony Morgan
Baba Piprani
Erik Proper
Gerhard Skagestein
Peter Spyns
Serge Valera
Theo van der Weide
Joost van Griethuysen
Jos Vos
Jan Pieter Wijbenga

SWWS 2011 Program Committee

Grigoris Antoniou
D. Bell
Elisa Bert
Christoph Bussler
John Debenham
Stefan Decker
Usama Fayyad
Avigdor Gal
Aldo Gangemi
Carole Goble
Nicola Guarino
Mohand-Said Hacid
Jiawei Han
Manfred Hauswirth
Mustafa Jarrar
Q. Li
F. Ling
Ling Liu
Robert A. Meersman
Mukesh Mohania
John Mylopoulos
Masoud Nikvesh
Pierre-Yves Schobbens
Amit Sheth
C. Sierra
Peter Spyns
Y. Sure
Katia Sycara
Zahir Tari
Robert Tolksdorf
Mihaela Ulieru
Susan Urban
Ramasamy Uthurusamy
Wil van der Aalst
Frank van Harmelen
Kyu-Young Whang
Kokou Yetongnon
Lotfi Zadeh
L. Zhou
H. Zhuge

MONET 2011 Program Committee

Kevin C. Almeroth
Frederic Andres
Russell Beale
Yiwei Cao
Tiziana Catarci
Richard Chbeir
Ioannis T. Christou
Karin Coninx
Simon Courtenage
Juan De Lara
Anna Formica
Nikos Komninos
C.-C. Jay Kuo
Peter Leijdekkers
Stephen Marsh
Rebecca Montanari
Michele Missikoff
Nuria Oliver
Marco Padula
Manish Parashar
Andrew Phippen
Nitendra Rajput
Tommo Reti
Ahmed M. Safwat
Nicola Santoro
Tim Strayer
Henri Ter Hofte
Thanassis Tiropanis
Yoshito Tobe
Riccardo Torlone
Mikael Wiberg
Adam Wojciechowski

SeDeS 2011 Program Committee

Saiful Akbar
Antonia Albani
Dimitris Askounis
Luke Chen
Johan Criel
Frode Eika Sandnes
José Fernán Martínez
Bin Guo
Stijn Hoppenbrouwers
Weihua Li
Yue Liu
Ying Liu
Peter Spyns
Zhiwen Yu
Jiehan Zhou

VADER 2011 Program Committee

Muhammad Ali Babar
Dharini Balasubramaniam
Javier Cámara
Carlos Canal
Sorana Cîmpan
Pedro J. Clemente
Cristóbal Costa-Soria
Carlos E. Cuesta
Laurence Duchien
José Luiz Fiadeiro
Howard Foster
Marcos López-Sanz
Zakia Kazi-Aoul
Nicole Lévy
Tommi Mikkonen
Juan Manuel Murillo
Elena Navarro
Amparo Navasa
Sascha Ossowski
Renaud Pawlak

Jennifer Pérez
Miguel Angel Pérez-Toledano
Dewayne E. Perry
Pascal Poizat
Isidro Ramos
Omer Rana
Pilar Romay
Gwen Salaün
Bradley Schmerl
Jean-Guy Schneider
Chantal Taconet
Antony Tang
Eddy Truyen
Juan Manuel Vara
Houman Younessi
Michel Wermelinger

Supporting and Sponsoring Institutions

OTM 2011 was proudly supported or sponsored by Vrije Universiteit Brussel in Belgium, Curtin University of Technology in Australia, Universidad Politecnica de Madrid in Spain, the Object Management Group, and Collibra.

Computing for Human Experience: Semantics Empowered Cyber-Physical, Social and Ubiquitous Computing beyond the Web

Amit Sheth

Kno.e.sis, Wright State University, USA

Short Bio

Amit Sheth is an educator, research and entrepreneur. He is the LexisNexis Ohio Eminent Scholar at the Wright State University, Dayton OH. He directs Kno.e.sis - the Ohio Center of Excellence in Knowledge-enabled Computing which works on topics in Semantic, Social, Sensor and Services computing over Web and in social-cyber-physical systems, with the goal of transitioning from information age to meaning age. Prof. Sheth is an IEEE fellow and is one of the highly cited authors in Computer Science (h-index = 67) and World Wide Web. He is EIC of ISI indexed Intl. Journal of Semantic Web & Information Systems (http://ijswis.org), is joint-EIC of Distributed & Parallel Databases, is series co-editor of two Springer book series, and serves on several editorial boards. By licensing his funded university research, he has also founded and managed two successful companies. Several commercial products and many operationally deployed applications have resulted from his R&D.

Talk

"Computing for Human Experience: Semantics empowered Cyber-Physical, Social and Ubiquitous Computing beyond the Web"

Traditionally, we had to artificially simplify the complexity and richness of the real world to constrained computer models and languages for more efficient computation. Today, devices, sensors, human-in-the-loop participation and social interactions enable something more than a "human instructs machine" paradigm. Web as a system for information sharing is being replaced by pervasive computing with mobile, social, sensor and devices dominated interactions. Correspondingly, computing is moving from targeted tasks focused on improving efficiency and productivity to a vastly richer context that support events and situational awareness, and enrich human experiences encompassing recognition of rich sets of relationships, events and situational awareness with spatio-temporal-thematic elements, and socio-cultural-behavioral facets. Such progress positions us for what I call an emerging era of computing for human experience (CHE). Four of the key enablers of CHE are: (a) bridging the physical/digital (cyber) divide, (b) elevating levels of abstractions and utilizing vast background knowledge to enable integration of machine and human perception, (c) convert raw data and observations, ranging from sensors to social media, into understanding of events and situations that are meaningful to humans, and (d) doing all of the above at massive scale covering the Web and pervasive computing supported humanity. Semantic Web (conceptual models/ontologies and background knowledge, annotations, and reasoning) techniques and technologies play a central role in important tasks such as building context, integrating online and offline interactions, and help enhance human experience in their natural environment.

Privacy and the Cloud

Siani Pearson

Hewlett-Packard Laboratories

Short Bio

Siani Pearson is a senior researcher in the Cloud and Security Research Lab (HP Labs Bristol, UK), HP's major European long term applied research centre. She has an MA in Mathematics and Philosophy from Oxford and a PhD in Artificial Intelligence from Edinburgh. She was a Fellow at the Computer Lab in Cambridge University, and for the last 17 years has worked at HP Labs in a variety of research and development programs including collaborations with HP business units and EU PRIME (Privacy and Identity Management for Europe) project.

Siani's current research focus is on privacy enhancing technologies, accountability and the cloud. She is a technical lead on regulatory compliance projects with HP Privacy Office and HP Enterprise Services, and on the collaborative TSB-funded EnCoRe (Ensuring Consent and Revocation) project.

Talk

"Privacy and the Cloud"

Cloud computing offers a huge potential both for efficiency and new business opportunities (especially in service composition), and is almost certain to deeply transform our IT. However, the convenience and efficiency of this approach comes with a range of potential privacy and security risks. Indeed, a key barrier to the widespread uptake of cloud computing is the lack of trust in clouds by potential customers. This concern is shared by experts: the European Network and Information Security Agency (ENISA)'s cloud computing risk assessment report states "loss of governance" as a top risk of cloud computing, and "data loss or leakages" is one of the top seven threats the Cloud Security Alliance (CSA) lists in its Top Threats to Cloud Computing report.

In this talk I will assess how privacy, security and trust issues occur in the context of cloud computing and explain how complementary regulatory, procedural and technical provisions can be used to help address these issues. In particular, accountability is likely to become a core concept in both the cloud and in new mechanisms that help increase trust in cloud computing. It is especially helpful for protecting sensitive or confidential information, enhancing consumer trust, clarifying the legal situation in cloud computing, and facilitating cross-border data transfers. I will also talk about some of the innovative technical solutions that we are developing in HP Labs to enhance privacy in the cloud.

The Social Compute Unit

Schahram Dustdar
Vienna University of Technology (TU Wien)

Short Bio

Schahram Dustdar (ACM Distinguished Scientist), is full Professor of Computer Science with a focus on Internet Technologies heading the Distributed Systems Group, Vienna University of Technology (TU Wien).

From 1999 - 2007 he worked as the co-founder and chief scientist of Caramba Labs Software AG in Vienna (acquired by Engineering NetWorld AG), a venture capital co-funded software company focused on software for collaborative processes in teams. He is Editor in Chief of Computing (Springer) and on the editorial board of IEEE Internet Computing, as well as author of some 300 publications.

Talk

“The Social Compute Unit”

Social computing is perceived mainly as a vehicle for establishing and maintaining social (private) relationships as well as utilizing political and social interests. Unsurprisingly, social computing lacks substantial adoption in enterprises. Clearly, collaborative computing is firmly established (as a niche), but no tight integration exists of social and collaborative computing approaches to facilitate mainstream problem solving in and between enterprises or teams of people. In this talk I will present a fresh look at this problem and examine how to integrate people in the form of human-based computing and software services into one composite system, which can be modeled, programmed, and instantiated on a large scale.

GENI - Global Environment for Network Innovations

Niky Riga

GENI Project Office, Raytheon BBN Technologies

Short Bio

Niky Riga is a Network Scientist at Raytheon BBN Technologies. Niky joined the GENI Project Office (GPO) in March 2010. As a member of GPO, Niky is responsible for supporting GENI experimenters in integrating and deploying their experiments as well as advocating their requirements to the rest of the GENI community.

Before joining the GPO, Niky worked on multiple innovative projects within the Network Research department of BBN. Her focus is on designing and prototyping pioneering transport services for Mobile Ad-hoc Networks, while her main goal is making innovative, research ideas practical and implementing them on real systems. She has successfully led various integration efforts. Niky earned a Diploma in Electrical and Computer Engineering at the National Technical University of Athens, and an MS degree in Computer Science at Boston University.

Talk

"GENI - Global Environment for Network Innovations"

The Global Environment for Network Innovations (GENI) is a suite of research infrastructure components rapidly taking shape in prototype form across the US. It is sponsored by the US National Science Foundation, with the goal of becoming the world's first laboratory environment for exploring future Internets at scale, promoting innovations in network science, security, technologies, services, and applications.

GENI allows academic and industrial researchers to perform a new class of experiments that tackle critically important issues in global communications networks such as (a) Science issues: we cannot currently understand or predict the behavior of complex, large-scale networks, (b) Innovation issues: we face substantial barriers to at-scale experimentation with new architectures, services, and technologies (c) Society issues: we increasingly rely on the Internet but are unsure that can we trust its security, privacy or resilience GENI is enabling researchers to explore these issues by running large-scale, well-instrumented, end-to-end experiments engaging substantial numbers of real users. These experiments may be fully compatible with today's Internet, variations or improvements on today's Internet protocols, or indeed radically novel, clean slate designs. The GENI project is now supporting such experiments across a mesoscale build-out through more than a dozen US campuses, two national backbones, and several regional networks. If this effort proves successful, it will provide a path toward more substantial build-out.

In this keynote presentation, she will introduce GENI through a couple of example use-cases, she will review the growing suite of infrastructure and evolving control framework. She will also present previous and current experiments running in GENI.

Table of Contents

On the Move Academy (OTMA) 2011

Posters of Ontologies, DataBases, and Applications of Semantics (ODBASE) 2011

ODBASE 2011 Posters

Workshop on Enterprise Integration, Interoperability and Networking with Information Centric Engineering and Cybertechnologies (EI2N+NSF ICE) 2011

Enterprise Integration and Semantics

Information Centric Engineering

Interoperability Case Studies

Workshop on Industrial Case Studies Program and Industrical and Business Applications of Semantic Web Technologies (ICSP+INBAST) 2011

Workshop on Information Systems in Distributed Environments (ISDE) 2011

Process Management in Distributed Information System Development

Distributed Information Systems: Implementation Issues

Distributed Information Systems Applications

Workshop on Fact-Oriented Modeling (ORM) 2011

Industrial Applications of Fact-Oriented Modeling

Data Warehouse Modeling

Extensions to Fact-Oriented Modeling

Model Validation Procedures

Schema Transformations and Mapping

Workshop on Semantic Web and Web Semantics (SWWS+MONET+SeDeS) 2011

Ontology Development, Deployment and Interoperability (ODDI)

Data Access and Efficient Computation (DAEC)

Efficient Information Processing, Exchange and Knowledge Synthesis Algorithms (EIPKS)

Mobile and Networking Technologies for Social Applications (MONET)

Semantic and Decision Support (SeDeS)

Workshop on Variability, Adaptation and Dynamism in Software Systems and Services (VADER) 2011

Variability in Software Architecture

Dynamic and Adaptive Architectures

The 8th OnTheMove Academy Organizers' Message

"A good beginning is half way to success" is what a Chinese proverb teaches. Hence, the OTM Academy offers our next-generation researchers far more than other doctoral symposiums usually do.

OTMA PhD students get the opportunity to publish in a highly reputed publication channel, namely, the Springer LNCS OTM workshops proceedings. The OTMA faculty members, who are well-respected researchers and practitioners, critically reflect on the students' work in a highly positive and inspiring atmosphere, so that the students can improve not only their research capacities but also their presentation and writing skills. OTMA participants also learn how to review scientific papers. And they enjoy ample possibilities to build and expand their professional network. This includes dedicated personal feedback and exclusive time of prominent experts on-site. In addition, thanks to an OTMA LinkedIn group, the students can stay in touch with all OTMA participants and interested researchers. And last but not least, an ECTS credit certificate rewards their hard work.

Crucial for the success of the OTM Academy is the commitment of the OTMA faculty members whom we sincerely thank:

- Erich J. Neuhold (University of Vienna, Austria), OTMA Dean
- Alfred Holl (University of Applied Sciences, Nuremberg, Germany)
- Maria Esther Vidal (Universidad Simon Bolivar, Caracas, Venezuela)
- Josefa Kumpfmller (Vienna, Austria), OTM Integrated Communications Chair

The OTM Academy submissions are reviewed by an international Program Committee of well-respected experts together covering a wide area of expertise. We gratefully thank them for their effort and time:

- Galia Angelova (Bulgarian Academy of Science, Sofia, Bulgary)
- Christoph Bussler (XTime Inc., USA)
- Paolo Ceravolo (Universit degli Studi di Milano, Milan, Italy)
- Philippe Cudr-Mauroux (Massachusetts Institute of Technology, USA)
- Jaime Delgado, (Universitat Politcnica de Catalunya, Barcelona, Spain)
- Ling Feng, (Tsinghua University, Beijing, China)
- Marcello Leida (Khalifa University, Abu Dhabi, United Arab Emirates)
- Frdric Le Moul (University of Lyon, Lyon, France)
- Herv Panetto (Nancy University, Nancy, France)
- Erik Proper, (Public Research Centre - Henri Tudor, Luxembourg)
- Adam Wierzbicki (Polish-Japanese Institute of IT, Warsaw, Poland)

This year, two submissions were accepted as regular papers and one as a short paper for inclusion in the proceedings. We hope that you will find the papers

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 1–2, 2011.
© Springer-Verlag Berlin Heidelberg 2011

of these upcoming researchers promising and inspiring for your own research activities. We also express our thanks to Christophe Debruyne (Vrije Universiteit Brussel – STAR Lab), who volunteered to be the OTMA 2011 unofficial "Twitter master".

August 2011

Peter Spyns
Anja Metzner

A Framework Proposal for Intelligent Management of Unexpected Exceptions in Workflow

Myriam Amina Derbali

University of Namur, 5000, Belgium
{myriam-amina.derbali}@fundp.ac.be

Abstract. Workflow management systems are recognized to provide essential and effective support to business process management. However, many events, either too infrequent or outside the business descriptions, are generally not taken into account and, consequently, are not integrated in the a priori definition of the workflow. This kind of exceptions are known as unexpected exceptions. Usually, handling them relies on human interventions that are potentially error prone and could take a lot of time. In this work, we propose a solution to manage unexpected exceptions. To that end, we aim at providing a framework that supports real time automatic management of these exceptions. It relies on the definition of a reaction to an exception that is guaranteed to be optimal and adapted to the workflow context. In particular, we use artificial intelligence techniques to generate the reactions automatically.

Keywords: exception management, unexpected exceptions, workflow, expert system.

1 Introduction

A workflow management system is a software system that provides workflow specification, interpretation mechanisms and workflow engine to support the workflow execution [4]. The workflow engine coordinates and schedules the tasks by invoking other software applications or human [6]. The workflow management systems(WfMS) have been successfully applied to support the execution of the repetitive and predictable workflows [4].

However, companies must change their workflow specification frequently in order to manage rapid changes due to dynamic nature of the marketplace or service environment [19,13]. Moreover, WfMS should support exception management and be able to dynamically adapt to changing environment. Some exceptions can be considered and handled by the designer during the design time of the workflow, they are known as expected exception [4]. Other exceptions cannot be foreseen before because they are raised by unanticipated events which are triggered by changes in the environment. This kind of exceptions is called unexpected exceptions. Usually, the management of this kind of exceptions requires

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 3–10, 2011.
© Springer-Verlag Berlin Heidelberg 2011

human intervention by using their knowledge and skills [10]. Unfortunately, these human intervention increase the cost of the management in terms of time.

To overcome these limitations, an **automatic support** for unexpected exceptions management avoiding human interactions as much as [14] should be provided. This automatic management should be **adapted to the context** [17] of the workflow execution (actors, environment, resources). In addition, the automatic management should propose an **optimized** reaction in order to minimize the being used resources and effective damages. Finally, the detection and the handling of unexpected exceptions should be performed **in real time** during the workflow execution.

Logistics workflows are typical examples of workflows with dynamic environment (weather factor, traffic jam, ...). In these worflows, the unexpected exceptions can be produced by several external resources during the execution time due to the mobility of the modelled activities. The logistics exceptions can have critical consequences, Becker argues in [2] that companies can loose between 9 per cent to 20 per cent of their share value over a six-month period due to logistics exceptions. For this reason, it is crucial to detect and recover exceptions appropriately at real time in order to minimise the exception consequences.

In our work, our objective is to propose a framework which allows a real time automatic management of unexpected exceptions. We will explore a framework based on an expert system. Our main contribution consists in proposals to produce the most optimized and adapted reaction to unexpected exceptions.

This paper is structured as it follows. We start Section 2 by presenting a background about the exception classification and management. Then, Section 3, we detail an overview of the works about the exceptions management. In Section 4 we expose our research questions. We describe a preliminary plan for addressing the identified problem in Section 5. Finally, Section 6 concludes our work.

2 Exception Management

The impetus for our research is *exception*. Exception is an abnormal situation that occurs during the execution of the workflow which can cause critical consequences. Exceptions may be different according their causes. In addition each type of exception could have a specific management. In this section we summarize the classification of exceptions presented in [4] and detail the steps of their management of the exceptions.

2.1 Exception Classification

According to the literature [4], exceptions are classified into three types: (1) *failures* corresponding to failures of application or platform which are not specific to the business process, (2) *expected exceptions* which correspond to predictable deviations from the normal behaviour of the workflow; and (3) *unexpected exceptions* corresponding to inconsistencies between the business process in the real world and its workflow representation. While, failures are generally handled at the system level, the expected and unexpected exceptions are specific to workflow management. For this reason, most of researches focus on their management. Let us clarify these two types of exceptions.

1. *Expected exceptions* are known in advance to the workflow designer, and they occur occasionally or periodically. The way of handling them can be completely decided at design-time. Casati distinguishes in [4] classes of expected exceptions by events that causes them (eg, the deadline of a task, modification in data, ...).
2. *Unexpected exceptions* are always external to the workflow system [12]. They arise from unpredictable changes in the environment, or some other factor that cannot be planned at design-time. It has been observed that exceptions occur rather frequently in real working environments [4,8]. Heinl in [10] argue that their detection and handling require human intervention.

2.2 Exception Management

To take these exceptions into account , whatever their type is, it is necessary that WfMS become able to handle them. This handling is commonly [7,15] structured in three steps:

1. **Detecting exception:** The exception in a workflow is captured by supervising the input, the output and the behavior of the workflow instance. When an exception is identified, it will be classified according to its type based on predefined patterns.
2. **Diagnosing exception:** The reaction that recovers the detected exception is selected from a set of predefined actions. This selection is done by considering the exception pattern and the current case of the workflow (data flow, control flow and the used resources).
3. **Recovering exception:** The recovery action is executed in order to drive the execution of the workflow towards a stable situation.

Note that, these three steps can be done either manually or automatically.

3 Related Work

Several initiatives have been conducted in order to study how exceptions can be managed in the WfMS [16,3]. Some of these studies, which focus on unexpected exceptions management, are discussed in this section. These works detect, diagnostic, and recover an unexpected exception in different ways.

Several works assume that the detection of an unexpected exception requires human intervention. For instance, [1] and [12] propose that a human administrator notifies the WfMS, when the unexpected exception occurs. However, the human detection increases the time of exception management process [18]. It is appropriate to detect and resolve exception in real-time in order to react to the exception whenever they occur. This can avoid critical consequences.

As explained above, once unexpected exception is identified, an appropriate reaction has to be selected. For this reason, several diagnostic and recovery strategies are proposed in the literature. For example, Casati et al. in [5] propose to handle unexpected exceptions by modifying the work specification. Consequently, all the next instances will be executed using the modified specification and that may cause WfMS to be more complex. Another unexpected exception diagnostic and recovery strategy is proposed by Adams in [1], called Worklet Service. This service offers a set of handling processes that contains a set of actions automatically applied. Actions, such as cancel, suspend, complete, fail, restart and compensate, can carry out at a workflow case or a task of workflow. The handling process can also include a workflow compensation. The selection of those handling exception processes is provided automatically using a decision trees and the current context. However, decision trees hardly support changes in data. This change could involve in the reconstruction of all trees and costs in time and resource. Moreover, there is only one reaction unfired by rules decision trees and it is not necessary the optimized one.

In summary, approaches to manage unexpected exception workflow usually relies on human intervention which directly impedes on the basic aim of workflow systems (to bring greater efficiencies to work practices). Another common theme is the lack of adaptation and optimisation of the exception reaction. This led us to opt for a real time automatic unexpected exceptions management. Our solution will be based on declarative expert system to provide an optimized and adapted reaction.

4 Research Questions

The main goal of our proposal is to develop an adapted, optimized and real-time unexpected exception management framework that improves workflow management system in terms of reliability. By reliability we mean the ability of the workflow management system to support the execution in normal situation as well as abnormal one. To that end, we identified three major research directions:

- How can we identify and catch unexpected exceptions (defined in 2.1) automatically at real time?
- How can we generate automatically an adapted recovery action that take into account the context of the workflow instance?
- How can we generate automatically an optimized recovery action that uses minimum possible resources?

5 Research Methodology

Based on the research questions defined in the previous section, we have identified the research methodology to achieve our goal. Our methodology is divided into three main stages: (1) theory elaboration, (2) prototype implementation, (3) and evaluation. The theory elaboration will be validated step by step by the implementation of the prototype. In addition, the evaluation will be made by using a real use case that will conclude the implementation. Let us detail each in turn.

5.1 Theory Elaboration

The first stage of our research methodology consists in defining of the proposed framework that allows the management of unexpected exceptions. Figure 1 shows an overview of the framework architecture that supports the workflow management life-cycle. This life-cycle begins from the specification of the workflow (design time), through the interpretation and execution of the workflow specification (runtime) until analyzing of the operational workflow performance (diagnostic time).

Our framework assumes that the workflow is already defined by an expert. For this reason, the framework deals, at design time, with an existing workflow. In this phase, the workflow can be redesigned as we will explain afterwards. At runtime, our framework uses a set of components that collaborate together in order to manage the unexpected exception whenever they occur. These set of components consist in: (1) a *workflow engine* that execute the workflow; (2) a *CEP* (complex event processing) engine that allows to detect the occurrence of an exception in real-time; (3) an *expert system* which is an interactive computer-based decision tool that uses both rules (facts) and conditions to solve difficult decision problems which are based on knowledge acquired from an expert. At diagnostic time, our framework analyses the historic of the occurred exception (log) in order to estimate whether the workflow specification should be redesigned to take into account frequently occurred exception.

To implement the framework, we divide our work into four important steps. The three first steps focus on the unexpected exception management (see Section 2) while, the last step focuses on analysis and reengineering.

- **Step 1: Detecting exception**
  We study how exceptions can be detected in real-time by using predefined patterns. For this reason, we plan to propose an exception model that allows to describe the semantic of the exception. This semantic specifies the elements and the information that an exception may contain (e.g. occurrence time, source, and priority level). We also plan to define, in this step, a classification method that allows to group exceptions in patterns by analysing the exception models.

- **Step 2: Diagnosing exception**
  Another related question concerns the use of an expert system for the exceptions diagnostic. Indeed, we assume that the selection of the recovery action can be considered as the result of the collaboration of the knowledge base and inference engine. We consider also that the inferred recovery action should be adapted to the context of the workflow instance as well as

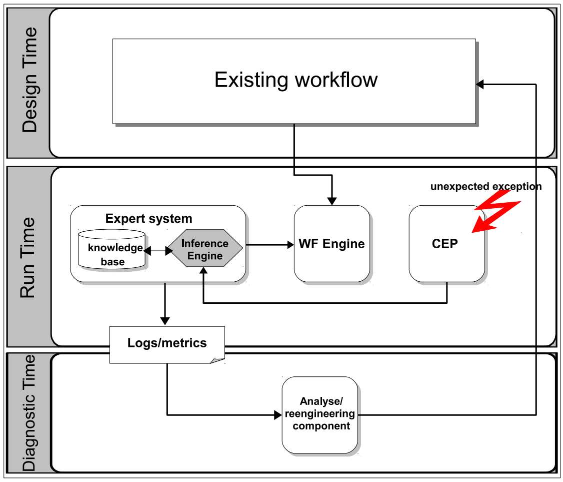


Fig. 1. Global architecture Framework

it should use optimal resources. To reach this end, we plan to define a rule based model that allows to specify the knowledge base. In addition, to verify the consistency of the knowledge base, a formal method should be defined. Moreover, a recovery inference algorithm is proposed. This algorithm allows the generation of a set of recovery actions that handles the effect of the exception. The recovery action generation is based on the knowledge base as well as a set of predefined adaptation and optimization rules. The generated recovery actions are considered as an alternative workflow (recovery workflow).

– **Step 3: Recovering exception**
We focus on the monitoring of the recovery workflow execution by determining whether the failed workflow instance that should be suspended or removed. For this reason, we plan to define, in this step, a method that allows a flexible execution of the workflow by deviating or evolving the instance of the failed workflow.

– **Step 4: Workflow reengineering**
This step explores how to analyse the occurred exceptions and inject the recovery action into the specification of the workflow if an exception occurs frequently. For this reason, we plan to define an exception occurrence threshold that allows to decide whether the exception needs to be taken into account or not in the specification workflow. In addition, we consider to propose a reengineering method that helps to redesign the workflow specification in order to take into account the recovery action of the frequently occurred exception.

5.2 Prototype Implementation

The second stage of our research methodology consists in the development of a prototype for our framework. This prototype will be made up of several components that collaborate together in order to manage workflow execution in a reliable way. The developed prototype relies on the YAWL WfMS [9]. YAWL is chosen for two main reasons. On one hand, the YAWL language is well-expressive since it supports the most Workflow patterns [11]. On the other hand, the YAWL engine is open source is used both in research and in industry. For this reason, this WfMS will be used to model and execute both workflow and the recovery workflow.

In order to detect the occurrence of an exception in real-time, we plan to extend the YAWL engine with a new developed CEP engine (complex event processing). Our framework will use a Prolog based expert system which manage the rule based knowledge base and infer recovery actions. The CEP engine listens to the operating environment of the workflow execution and detect any event that may affect its smooth running. If an exception is detected at run time, the worflow should be interrupted in order to offer a recovery action to handle the problem as much as possible. Decisions concerning the management of the interrupted workflow as well as the definition of the recovery action will be generated by the expert system on base of the current state of workflow, information provided by the CEP and any useful information obtained through other agents in the environment. Once the solution is generated, it will be injected into Yawl engine.

In addition, YAWL will be extended with an analysis component that allows to parse the history of occurred exceptions and estimate whether the recovery of an exception needs to be injected into the worflow specification. This analysis component will collaborate with a reengineering component that allows to redesign the workflow in order to take into account the workflow recovery as explained above.

5.3 Evaluation

The third stage of our research methodology consists in the evaluation of our framework. To that end, we plan to use a real logistics workflow as the validation use case. This use case is chosen because logistics workflows are usually executed in dynamic environments (weather factor, traffic jam, ...). So that, several unexpected exceptions can arise from many external resources at execution time.

6 Conclusion

In this paper we have outlined the project proposal for an exceptions management framework. This framework favours real time automatic management of unexpected exceptions. To achieve this aim, we propose the use artificial intelligence techniques to generate the reactions automatically.

Acknowledgment. This work is supported by the Walloon Region.

References

1. Adams, M.: Facilitating dynamic flexibility and exception handling for workflows. Ph.D. thesis, Queensland University of Technology, Australia (2007)
2. Becker, T.: Putting a price on supply chain problems: study links supply chain glitches with falling stock price (2000), http://gtresearchnews.gatech.edu/newsrelease/CHAINR.html
3. Borgida, A., Murata, T.: Tolerating exceptions in workflows: a unified framework for data and processes. In: Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration, WACC 1999, pp. 59–68. ACM, New York (1999)
4. Casati, F.: A Discussion on Approaches to Handling Exceptions in Workflows. In: Proceedings of the CSCW Workshop on Adaptive Workflow Systems, Seattle, USA (1998)
5. Casati, F., Ceri, S., Pernici, B., Pozzi, G.: Workflow evolution. Data and Knowlledge Engeneering 24, 211–238 (1998)
6. Coalition, W.M.: Workflow management coalition terminology glossary (1999)
7. Dellarocas, C., Klein, M.: A knowledge-based approach for handling exceptions in business processes. Journal of Computer-supported Collaborative Work. Special Issue on Adaptive Workflow Systems, 399–412 (2000)
8. Eder, J., Liebhart, W.: The workflow activity model wamo. In: Proceedings of the 3rd International Conference on Cooperative Information Systems (CoopIs), pp. 87–98 (1995)
9. Foundation, Y.: Voir la page, http://www.yawlfoundation.org/
10. Heinl, P.: Exceptions during workflow execution. In: Proceedings of the Sixth International Conference on Extending Database Technology, Workshop on Workflow Management, Valencia, Spain (1998)
11. Initiative, W.P.: Voir la page, http://www.workflowpatterns.com/
12. Mourao, H.: Supporting effective unexpected exception handling in workflow management systems within organizational contexts. Ph.D. thesis, University of Lisbon (2007)
13. Reichert, M., Dadam, P.: A framework for dynamic changes in workflow management systems. In: Proceedings of Eighth International Workshop on Database and Expert Systems Applications, pp. 42–48 (1997)
14. Rinderle-Ma, S., Reichert, M.: Data–driven Process Control and Exception Handling in Process Management Systems. In: Martinez, F.H., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 273–287. Springer, Heidelberg (2006)
15. Sadiq, S., Orlowska, M.: On capturing exceptions in workflow process models. In: Business Information Systems (2000)
16. Song, Y., Han, D.: Exception Specification and Handling in Workflow Systems. In: Zhou, X., Zhang, Y., Orlowska, M.E. (eds.) APWeb 2003. LNCS, vol. 2642, pp. 495–506. Springer, Heidelberg (2003)
17. Soylu, A., Causmaecker, P., Desmet, P.: Context and adaptivity in pervasive computing environments: Links with software engineering and ontological engineering. Journal of Software 4(9) (2009)
18. Strong, D., Miller, S.: Exceptions and exception handling in computerized information processes. ACM Trans. Inf. Syst. 13, 206–233 (1995)
19. Ter Hofstede, A.M., Van Der Aalst, W.M.P., Adams, M., Russell, N. (eds.): Modern Business Process Automation: YAWL and its Support Environment. Springer, Heidelberg (2010)

Towards Designing Enterprises for Evolvability Based on Fundamental Engineering Concepts

Peter De Bruyn

University of Antwerp,
Faculty of Applied Economics,
Department of Management Information Systems
{peter.debruyn}@ua.ac.be

Abstract. Contemporary organizations are operating in a hypercompetitive environment, in which they are faced with challenges such as increasing complexity and increasing change in many or all of their aspects. As such, current organizations need to be more agile to keep up with the swiftly changing business environment. However, current development practices for information systems supporting these organizations appear to be insufficient to deal with these levels of changing complexity. Normalized Systems theory therefore proposes a theoretical framework that explains why current modular structures in information systems are intrinsically limited in terms of evolvability, as well as how modular structures can be built without these limitations, thus exhibiting evolvable modularity. The goal of the proposed Ph.D. research project is to develop a contribution to how evolvable modularity as a theoretical framework can be further extended from the technological level to the business level and as such support business processes, enterprise architectures and their supporting IT systems. The ultimate purpose will be to make organizations as a whole more agile by developing so-called normalized design patterns (domain models) on the business level in order to allow a more deterministic approach enabling the expanding of enterprises.

Keywords: Normalized Systems, Enterprise Engineering, agility, evolvability, design patterns, domain models.

1 Introduction: The Challenge of Changing Complexity

Contemporary organizations are operating in hypercompetitive environments, forced to constantly monitor their environment for new business opportunities and striving for customer satisfaction by delivering products and services of unprecedented quality. This has resulted in enterprises that are faced with challenges such as *increasing complexity* and *increasing change* in many or all of their aspects, including their products and services, business processes, business rules and organizational structure. As such, current organizations need to be more agile to keep up with the swiftly changing business environment, while

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 11–20, 2011.
© Springer-Verlag Berlin Heidelberg 2011

managing changing complexity both at the enterprise and the information systems level [23]. Indeed, both IS researchers [32] and practitioners [19] stress the need of being able to address both agility and the complex alignment between business and IT concurrently.

However, current development practices and methodologies for information systems supporting these organizations (typical structured development methodologies, object-oriented methodologies, etc.) appear to be insufficient to deal with this changing complexity. Some clear symptoms include growing IT departments, failing IT projects, and rising IT (maintenance) costs [18,31]. Also on the business level, organizations regularly seem to have difficulties in quickly adapting their business processes, organizational structure and 'operating procedures' [1,8] and a theoretically founded, deterministic approach to designing organizations for evolvability seems to be lacking. First, the available methodologies frequently seem to pay too little attention to large-scale aspects of real implementations, know only a very low adoption rate and generally constitute only a heterogeneous collection without clear integration [5,6,15,27]. Next, many of the available 'tools' are merely descriptive (i.e., notations) and not prescriptive, such as, for instance, the UML Activity Diagrams or the BPMN standard. Also within the field of enterprise architectures, more prescriptive approaches are called for [11].

Consequently, it is clear that in order to be finally able to 'engineer' enterprises and deal with the ever increasing changing complexity, a more deterministic approach regarding the design of organizations is required, called *Enterprise Engineering*. This approach is based on some already existing theoretical frameworks like organizational theory, business process management and enterprise modeling and tries to employ fundamental, proven and solid engineering principles in both theory and practice regarding the design of organizations.

In this respect, *Normalized Systems theory (NS)* has recently been developed, providing a theoretical framework allowing the development of ex-ante proven evolvable information systems, based on typical engineering concepts such as modularity, stability and entropy [21,22,23]. So far this theory has shown its value both in theory and in practice, mainly at the technical level. As such, this paper proposes a research approach that aims at extending and applying this existing theory on evolvable modularity to the enterprise level of business processes and enterprise architectures. In doing so, the research will aim to develop so-called normalized design patterns (domain models), using proven principles to design enterprises for evolvability. By using the same theoretical background on evolvability for both the technical and the organizational level (thus allowing traceability), it seems more realistic to make a contribution towards the identified need of tackling agility and business/IT alignment concurrently [32].

In the remainder of this paper we will first clarify the specific research motivation and objectives. Next, the related work and theoretical background will be explained in Section 3. Afterwards, the proposed research methodologies and activities will be highlighted. We will end this paper with some final conclusions and anticipated contributions.

2 Research Motivation and Objectives

Section 1 coined today's business environment as confronted with 'changing complexity', applicable at both the business and the IT level. As will be explained in Section 3, NS theory has been suggested to engineer evolvable information systems by enabling the IT infrastructure to quickly adapt to changing business requirements [21,22,23]. Thus, this theory can solve the increasing complexity issue at the software level.

The question then becomes how to tackle this changing complexity at the organizational level and how to map in an unambiguous way organizational requirements into the derived NS software elements. Consequently, the main goal of this research project is to develop a contribution to *how evolvable modularity as a theoretical framework can support business processes and enterprise architectures at a business level while achieving maximum alignment (traceability) between the organization and its supporting IT systems* by using evolvable modularity as the theoretical foundation on both levels. More specifically, the *research objectives* can be formulated as:

1. To create a *methodology* describing how requirements for information systems based on business processes and enterprise architectures can be transformed with high levels of alignment/traceability to NS;
2. To create *normalized design patterns* or domain models using and/or enabling the above methodology, i.e., normalized business processes and architectures as kind of 'organizational building blocks' to 'expand' organizations.

Moreover, considering both the organizational and technical level from the same theoretical perspective is, to our knowledge, new.

3 Related Work

In this section we will first briefly explain the essence of NS. Next, we argue how NS theory could be further extended from a technical to an organizational level.

3.1 Normalized Systems Theory

Most methodologies for building information systems use the systems theoretic notion of modularity. However, indications are given that the evolvability of these modular structures is limited. For instance, *Manny Lehman's Law of increasing complexity* stated that as software changes over time, its structure tends to become more and more complex, reflecting structure deterioration or degradation, and resulting in ever less evolvable information systems [17]. From the perspective of an individual change to the system, this essentially means that the cost of this individual change to the system will be ever increasing. In a similar way, Brooks stated that software maintenance is an '*entropy*-increasing' process (i.e., resulting in increasing disorder) [2], just as Lehman himself proposed his law initially as an instantiation of this second law of thermodynamics [16].

NS theory tries to tackle these issues by first assuming an *unlimited systems evolution* (i.e., time T $\rightarrow \infty$ in which the number of primitives and their dependencies becomes unbounded) and next demanding that every change to the system (e.g., adding a data attribute) should exhibit *stability* [21,22,23]. Stability originates from systems theory, demanding that a bounded (i.e., finite) input function should result in bounded output values, even as T $\rightarrow \infty$. For information systems, this means that a bounded set of changes (e.g., adding a data attribute) should result in a bounded amount of impacts to the system, even for T $\rightarrow \infty$. In other words, stability demands that the impact of a change is only dependent on the nature of the change itself. Those changes having impacts that are also dependent on the size of the system, are called *combinatorial effects* and should be eliminated from the system structure in order to attain stability.

Consequently, information systems that exhibit stability are defined as *Normalized Systems* [21,22,23]. It has been shown that in order to develop such NS systems without combinatorial effects, the following four design *principles* are to be adhered to at all times, as a necessary but not a sufficient condition [21,22,23]:

- *Separation of Concerns (SoC)*, requiring that every change driver or concern is separated from other concerns;
- *Data Version Transparency (DvT)*, requiring that data is communicated in version transparent ways between components;
- *Action Version Transparency (AvT)*, requiring that a component can be upgraded without impacting the calling components;
- *Separation of States (SoS)*, requiring that each step in a workflow is seperated from the others in time by keeping state after every step.

Applying these principles results in a very fine-grained modular structure (i.e., consisting of large numbers of small modules), having its own 'design' complexity. Therefore, the encapsulation of software constructs into five higher-level structures, called *elements*, was proposed. These form a kind of 'building blocks' together offering the core functionality of information systems: data elements, action elements, workflow elements, trigger elements and connector elements. Applications based on these elements are then considered as an aggregation of a number of instances of these elements. As these elements are proven to be free of combinatorial effects, also the applications based on them are free of combinatorial effects [21,22,23].

The NS rationale explained above is visually represented in Figure 1. First, NS theory focuses on the use of modular structures within a software system (e.g., objects, classes, structures, etc.), which generally result in combinatorial effects when exposed to change. Further, NS has proposed 4 principles in order to avoid those combinatorial effects in a proven way resulting in a very fine-grained modular structure. Finally, 5 elements or patterns have been suggested on the software level which are instantiated throughout the NS application enabling this fine-grained structure. We will now argue how this theory can be extended from the technical level to the organizational level.

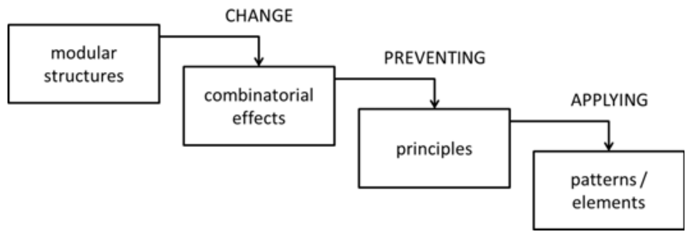


Fig. 1. The general rationale behind the NS theory

3.2 Extending NS Theory to the Organizational Level

In this research project, it will be investigated how evolvable modularity can be applied at the enterprise level, both related to business processes and enterprise architectures. In order to study the enterprise level, the use of BPMN as a business process modeling language [29] and DEMO as an an enterprise architecture methodology [5] will be investigated. A first goal of the research project will be dedicated into how enterprise architectures and business processes can be unambiguously be mapped onto NS elements. However, the scope of this project is not limited to a translation issue from enterprise to software level, as translating non-normalized enterprises to normalized software would clearly lead to suboptimal results. As such, organizations themselves should also be normalized to some extent, if the NS rationale is indeed applicable at this enterprise level.

The first question that needs to be answered is whether it is reasonable to consider enterprises as *modular structures*. Regarding this aspect, Campagnolo and Camuffo [3] studied the concept of modularity in management studies and identified 125 studies on modularity in management literature in the period 1986-2007, stating that "both theoretical and empirical studies on this topic" have flourished in product design, production systems and organizational design issues. For instance, Op't Land [25] investigated the splitting and merging of organizations from an enterprise design viewpoint ending up with construction rules using typical constructs such as change, cohesion and coupling between units and thus supporting the claim of considering organizations as modular systems.

Next, also concepts such as *stability*, *entropy* and *combinatorial effects* seem to correlate well with known phenomena at the business level. For example, Janow used a Shannon entropy based approach to explain organizational inertia and the slowdown of organizational decision-making [14]. The existence of combinatorial effects at the organizational level can easily be demonstrated by means of a straightforward example: consider for instance the situation in which an organization has N organizational actors using P business processes which all include some kind of 'message' task to other (internal or external) stakeholders. Suppose that this task is not separated from the other tasks in those business processes and that one day, a new business rule demands that the way of sending messages has to change (e.g., by adopting a new structure or heading for sending letters, using a different protocol to send e-mails, changing the overall communication policy, . . .). Enforcing this rule will clearly generate a combinatorial effect. First, the required effort to implement the change will depend on the number of processes or tasks using the message functionality and thus will not satisfy the requirement of 'stability' as it depends on the overall size of the organization. Second, the change will require white-box inspection of every single process and task, as it is not known upfront where all the scattered-around message functionality is present and should be changed. Applying the *SoC* principle of NS (i.e., encapsulating the task in its own construct) could prevent the instability.

Finally, also the use of recurring *patterns* to 'expand' organizations has been gaining attention during the past decade. Some of these approaches primarily concentrate on 'generic' data-aspects of organizations (e.g., [7,9,30]), processes (e.g., [20]) or both (e.g., [28]). Also DEMO [5] and REA (Resources, Event, Agents) [12] consider organizations as an aggregation of instances of one generic pattern and apply in that way a somewhat similar reasoning as NS. However, none of these frameworks explicitly addresses combinatorial effects or other evolvability criteria, frequently lack adherence to prescriptive design principles [4] and mostly lack integration with their technical implementation [15].

As such, we can refer again to Figure 1: throughout this section we have tried to demonstrate (although very briefly due to the limited space available) how each of NS rationale aspects apply at the enterprise level. Also, some preliminary principles on the business level based on NS theory have already been proposed [13,33,34], although no realistic large-scale case studies have been performed yet. In conclusion we can state that similar approaches to NS have been previously proposed at the enterprise level, but further research is definitely required.

4 Research Methodology and Anticipated Research Activities

Given the above mentioned research objectives, it is clear that primarily a *Design Science Research methodology* is suited to provide the required research setting and guidance. Indeed, as for example [10,24,26] mention, design science research is aimed at solving identified relevant organizational problems by developing and testing (useful) artefacts (i.e., "creating things") rather than developing and testing theories that try to explain organizational and human phenomena (i.e., "understanding things"). The artefacts should be purposeful in that they address a heretofore unsolved problem or try to tackle the problem in a more efficient or effective way [10]. Applied to our research project, we will try to tackle the changing complexity issue by developing a prescriptive mapping methodology and normalized design patterns to enable the expanding of enterprises, based on the NS principles. Our approach is innovative, in the sense that we will use one theoretical foundation on both the technical and the business level and will prescriptively design enterprises primarily from the viewpoint of evolvability.

Applying the classification of March and Smith [24] our aim will be to build and evaluate both a method (i.e., guidelines) and models (i.e., representations).

On one hand, the methodology to map business processes and enterprise architectures to NS can be regarded as a method describing guidelines and even rigorous practices. On the other hand, the developed normalized design patterns (domain models) can be interpreted as models of how things are or should be.

Further elaborating on the *research activities*, design science is generally conceived as an iterative process of building and evaluating the aimed artefacts. However, Peffers et al. [26] use a more extensive Design Science Research Process, which we will adopt here. This process model is also depicted in Figure 2, where the six recommended steps for correctly executing design research are applied to the research project at hand.

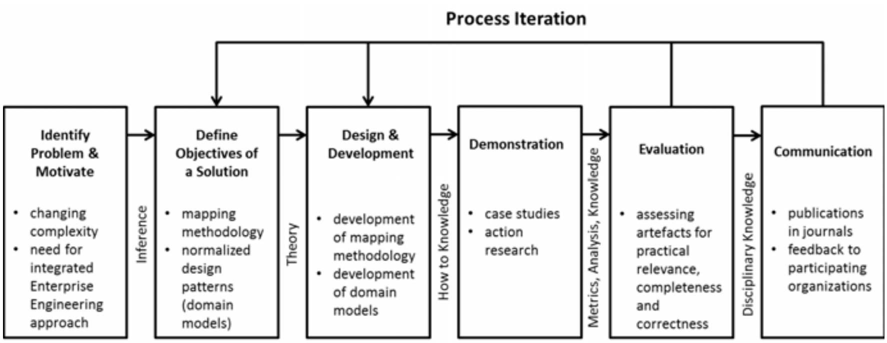


Fig. 2. The Design Science Research Methodology Process Model of Peffers et al. [26] applied to the research project

The problem identification and motivation as well as the aimed objectives of a solution were already extensively discussed in Sections 1 and 2. The *motivation* of the research is the need to create more agile organizations which leads to the *objective* of an integrated Enterprise Engineering approach (mapping procedure and design patterns) based on the NS theory.

Currently, the main efforts of the research project have been invested in conducting a thorough literature study of existing approaches regarding analysis and business process patterns, further supporting the problem formulation and motivation. As such, a first paper discussing the limited applicability of one specific business process pattern framework (i.e., the MIT Process Handbook [20]) to NS, due to (1) underspecification (2) no applied design principles and (3) top-down modeling, has been accepted for publication and presentation at an IARIA conference [4]. The next phases then mainly describe anticipated research.

During the *design & development* phase, both aimed artefacts will be jointly constructed. First, a number of case studies will be performed at different external organizations. Ideally, these organizations will be purposefully sampled, controlling for different industries and organizational dimensions. At this moment, a first case study has been initiated and information gathering has started. Next, based on the existing knowledge in the research group, mapping to NS will be performed and iteratively improved, starting from informal rules of thumb and informal principles into a more generic generic methodology to convert BPMN and DEMO into NS and investigate their traceability. While some initial efforts regarding this mapping have already been performed [34], this mapping has not been tested on realistic, medium- to large-scale examples.

Moreover, we anticipate it to be conceivable that during the development of the mapping procedure evidence arises that the mapping procedure will become more realistic when employing so-called elements or patterns at this business level, conceptually similar to those at the software level. Hence, parallel to the development of the mapping methodology, research will be aimed at developing normalized design patterns of business processes and enterprise architectures. These normalized patterns would constitute so-called domain models, but would be much more evolvable, fine-grained and reusable than its current predecessors. For instance, ‘payment’ or ‘send message’ processes could turn out to be basic patterns. A catalogue of reusable and evolvable artefacts can then be built, allowing the bottum up aggregation towards evolvable (parts of) organizations.

Next, concerning the *demonstration & evaluation* of the constructed artefacts, a multi-faceted evaluation will be aimed for, based on evaluation methods provided by [10] and [26]. First, during each research cycle, the constructed mapping procedure will be tested and altered to better suit the research objective of enhancing determinism and delivering unambiguous guidelines by means of *case study research*. That way, gradually studying more complex and real-life organization settings should ameliorate the generalizability of the mapping method. The generated domain models will be tested for completeness, internal consistency and fidelity with real world phenomena. Further, it has to be verified whether the domain models indeed exhibit stability, i.e., the absence of combinatorial effects. Second, the artefacts will be applied for making recommendations in real-life cases, which could be regarded as *action research* as the researcher will actively participate (i.e., suggesting recommendations) in the case. The effects in practice as well as practitioners feedback, will be studied and will deliver practical feedback and improvements to incorporate in later research cycles.

Finally, regarding the *communication* phase, the publication of the research results in international journals and conference proceedings is aimed for. Also, it will be investigated how the results can be integrated into other existing methodologies in widespread use (e.g., IBM’s Unified Process), in order to enable organizations to adopt more easily the research results of this project.

5 Conclusions and Anticipated Contributions

This paper started with considering ‘changing complexity’ as the core problem with which organizations are confronted these days. So far, NS theory has proven its value on the software level and seems to offer realistic opportunities to be extended onto the organizational level of business processes and enterprise architectures, which will be the main focus of this research project. Hence, the final research contributions can be divided into theoretical and practical ones.

Regarding the theoretical contributions, first, efforts were made in the past to link the managerial level to classic, non-normalized software in a traceable way. Given the described situation of increasing complexity, it is almost impossible to achieve this in a dynamic environment due to increasing entropy. NS offers traceability between software architecture and the generated code, proven evolvability and controlled entropy. This way, it seems more realistic to achieve alignment between the managerial level and the software architecture. Second, in the past, business/ICT alignment was hampered by using different theoretical foundations at the organizational level (e.g., value management) and the software level (which is, in essence, a modular structure). Considering both levels from the same theoretical perspective (i.e., evolvable modularity) is new, and thus enables to focus concurrently on agility and alignment [32].

Regarding practical contributions, the valorization potential is particularly broad in the sense that the NS framework is completely independent from any specific technology or vendor. As such, a very wide spectrum of companies could benefit from the research outcomes: whether manufacturing or services organizations, companies developing or outsourcing their software applications, etc.

Acknowledgments. This research is funded by a Research Grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

References

1. Beer, M., Nohria, N.: Cracking the code of change. Harvard Business Review 78(3), 13–23 (2000)
2. Brooks, F.P.: The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley (1975)
3. Campagnolo, D., Camuffo, A.: The concept of modularity within the management studies: a literature review. International Journal of Management Reviews 12(3), 259–283 (2009)
4. De Bruyn, P., Van Nuffel, D., Huysmans, P., Mannaert, H.: Towards functional and constructional perspectives on business process patterns. In: Proceedings of the Sixth International Conference on Software Engineering Advances, ICSEA (in press, 2011)
5. Dietz, J.L.: Enterprise Ontology: Theory and Methodology. Springer, Heidelberg (2006)
6. Fitzgerald, B.: An empirical investigation into the adoption of systems development methodologies. Information & Management 34(6), 317–328 (1998)
7. Fowler, M.: Analysis Patterns: Reusable Object Models. Addison-Wesley Professional (1996)
8. Hammer, M., Champy, J.: Reengineering the corporation: a manifesto for business revolution. Harper Business (1993)
9. Hay, D.C.: Data Model Patterns: Conventions of Thought. Dorset House (1995)
10. Hevner, A.R., March, S.T., Park, J.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
11. Hoogervorst, J.A.: Enterprise Governance and Enterprise Engineering. Springer, Heidelberg (2009)

12. Hruby, P., Kiehn, J., Scheller, C.: Model-Driven Design Using Business Patterns. Springer, Heidelberg (2006)
13. Huysmans, P., Bellens, D., Van Nuffel, D., Ven, K.: Aligning the Constructs of Enterprise Ontology and Normalized Systems. In: Albani, A., Dietz, J.L.G. (eds.) CIAO! 2010. LNBIP, vol. 49, pp. 1–15. Springer, Heidelberg (2010)
14. Janow, R.: Shannon entropy and productivity: Why big organizations can seem stupid. Journal of the Washington Academy of Sciences 90(1) (2004)
15. Kodaganallur, V., Shim, S.: Analysis patterns: A taxonomy and its implications. Information Systems Management 23(3), 52–61 (2006)
16. Lehman, M.: Programs, life cycles, and laws of software evolution. Proceedings of the IEEE 68(9), 1060–1076 (1980)
17. Lehman, M., Ramil, J.: Rules and tools for software evolution planning and management. Annals of Software Engineering 11, 15–44 (2001)
18. Lientz, B.P., Swanson, E.B.: Problems in application software maintenance. Communications of the ACM 24, 763–769 (1981)
19. Luftman, J., Ben-Zvi, T.: Key issues for it executives 2010: Judicious it investments continue post-recession. MIS Quarterly Executive 9(4), 263–273 (2010)
20. Malone, T., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C., Bernstein, A., Herman, G., Klein, M., O'Donnell, E.: Tools for inventing organizations: Toward a handbook of organizational processes. Management Science 45(3), 425–443 (1999)
21. Mannaert, H., Verelst, J., Ven, K.: The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability. Science of Computer Programming 76(12), 1210–1222 (2010)
22. Mannaert, H., Verelst, J., Ven, K.: Towards evolvable software architectures based on systems theoretic stability. Software Practice and Experience Early View (2011)
23. Mannaert, H., Verelst, J.: Normalized Systems: Re-creating Information Technology Based on Laws for Software Evolvability. Koppa (2009)
24. March, S.T.M., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
25. Op't Land, M.: Applying Architecture and Ontology to the Splitting and Allying of Enterprises. Ph.D. thesis, Technical University of Delft, NL (2008)
26. Peffers, K., Tuunanen, T., Rotherberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 45–77 (2008)
27. Riemenschneider, C., Hardgrave, B., Davis, F.: Explaining software developer acceptance of methodologies: A comparison of five theoretical models. IEEE Transactions on Software Engineering 28(12), 1135–1145 (2002)
28. Scheer, A.W.: Business Process Engineering – Reference Models for Industrial Enterprises. Springer, Heidelberg (1998)
29. Silver, B.: BPMN Method & Style. Cody-Cassidy Press (2009)
30. Silverston, L.: The Data Model Resource Book. A Library of Universal Data Models for All Enterprises, vol. 1. John Wiley & Sons (2001)
31. Standish Group: The standish group report: Chaos. Tech. rep. (1994)
32. Tallon, P.P., Pinsonneault, A.: Competing perspectives on the link between strategic information technology alignment and organizational agility: Insights from a mediation model. MIS Quarterly 35(2), 463–486 (2011)
33. Van Nuffel, D.: Towards Designing Modular and Evolvable Business Processes. Ph.D. thesis, University of Antwerp (2011)
34. Van Nuffel, D., Mannaert, H., De Backer, C., Verelst, J.: Towards a deterministic business process modelling method based on normalized theory. International Journal on Advances in Software 3(1/2), 54–69 (2010)

An Application Framework for Open Application Development and Distribution in Pervasive Display Networks

(Short Paper)

Constantin Taivan and Rui José

University of Minho, Department of Information Systems, Guimarães, Portugal
{constantin,rui}@dsi.uminho.pt

Abstract. Displays technology experienced a considerable progress and screens are increasingly pervasive in public spaces. Most of the display systems exist as singular non-interactive units showing static images, power point presentations or product videos. Further developments will consider the possibility of connecting the displays towards the Pervasive Display Networks. This approach will provide the background for highly interactive applications, thus creating a novel and powerful global communication medium.

The research activity proposed in this paper will consider the challenges raised by the need to support third-party application development and distribution. Consequently, the work will conceive and evaluate a web-based framework for the development, distribution and use of display-centred applications in Pervasive Display Networks.

Keywords: Display-centred applications, large scale display networks, ubiquitous computing and urban computing.

1 Introduction

The concept of Pervasive Display Networks (PDN) embodies the idea of creating large scale networks of autonomous displays that support integrated services. By removing the single management control that prevails in existing systems, PDNs should become open to content and applications from various sources and provide the ground for the emergence of a new communication medium for public places.

1.1 Motivation

A key enabler for PDNs is the ability to allow any third-parties to generate and publish applications for being consumed by displays in the network. An open model for the development of new applications is the key for the evolution of the network and its appropriation to new and unexpected uses, allowing multiple entities anywhere in a global network to become co-creators of value by developing new applications and content for themselves or for the community.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 21–25, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Numerous recent examples such as the Apple's App Store and Facebook demonstrate the power of enabling this model of third party applications to be easily created and deployed. Our vision is thus to move away from a world of closed display networks with content driven by advertisers and pushed to the public irrespective of their wishes to a new type of display systems, in which the respective applications may become an integral part of everyday life and a source of innovation within the communities in which they are situated [1].

1.2 Thesis

This research aims to define the set of abstractions and frameworks that may be needed to allow third-parties to generate and publish content and applications into PDNs. The topic is part of the objectives of the PD-NET project that provides the background for the proposal [2]. Globally, the thesis is going to assess the idea that PDNs can allow personalized, interactive, context-aware content for public displays to become a commodity and generate a community in which content produced by multiple entities is consumed by multiple displays.

1.3 Challenges

There are several sorts of challenges deriving from this research statement, but the most significant ones can be arranged in the following groups.

Execution environment. The concept of application is inherently linked with the concept of an execution environment. An execution environment may be defined as a collection of resources in which a task can execute, e.g., a Unix or Windows host, a J2EE environment. Within the context of displays networks and related sensors, an execution environment may also involve computational issues, such as the execution of code on the player nodes or on servers. Additionally, there is also the need to define and characterize the type of abstractions and resources that will be available to applications to represent the display network as a whole and the specific setting of each display. Specific challenges such as security and performance need also be considered to identify any particularities that are not addressed by existing solutions.

Content injection. The content generated by applications is not likely to be used individually. It will normally be consumed as a part of a broader context in which content from multiple applications is integrated into a specific display experience. Multiple content injection challenges arise from this need to support the coordinated presentation of content from multiple sources. Here, the challenges can further include scheduling, content representation and capabilities negotiation. The specific issues associated with scalability and the potential need to support substantial growth in the number of clients and updates should also be considered.

Situatedness. Situatedness is the ability of an application to adapt its behavior to the circumstances of the social setting in which it is being used. Many display-based applications may need to support this ability to sense situation and exhibit a situated behavior. Each application should have a mechanism allowing to configure and to adapt its behavior to certain local environment characteristics, users' profiles or specific usage sessions established before running. These specific issues include the need to represent situations, the need to acquire information from social setting, i.e. people, and the need to balance this with privacy.

Application Distribution. An application store represents an application distribution platform developed for publishing third-party applications developed everywhere in a community. Applying the concept of App Store in the context of public display networks, would open display networks to new applications from a wide range of sources, but raises significant challenges given the differences between the personal mobile device market and that of public display networks.

A key difference with many implications is the existence of multiple stakeholders in public display networks. While in a personal mobile device, a single owner decides what to install, the stakeholders in a display network may include display owners, venue owners, viewers (users) and content providers (application developers). Therefore existing models, while providing an inspiring model, cannot be applied.

Support for Multi-Screen Coordination. A unique feature of open public display systems is their ability to create experiences that span across multiple displays in a coordinated fashion. It should be possible for applications to address the multiple displays in an integrated network, based for example of location or proximity criteria. This type of fine-grained coordination will need new protocols that can enable large-scale multi-screen experiences through third-party application deployment.

Interaction. Public interactive displays lack efficient and clear abstractions for incorporating interactivity. There are no widely accepted frameworks on how to integrate interaction into the content presented on displays. While creating a specific interactive solution in itself is not a major challenge, the lack of generalized interaction abstractions represents a major obstacle to the widespread adoption of interactive features in public display. A key requirement towards the emergence of interactive applications for public displays will thus be a generic solution to the issue of how to distribute an interactive application over an open network of public displays and map it to whatever interactive features may be available on each of the sites in which the application may be used.

2 Related Work

Allowing third-party application development and distribution in large scale display networks represent a new topic in the field. Its originality derives from the emerging PDN. Currently, the main research effort in this area is given by the PD-NET project, which aims to explore and establish the foundations for an innovative communication medium based on large scale networked displays [2].

The state of the art in public display networks reports about small and medium display systems developed mainly for commercial purposes such as Times Square or social entertainment and various information broadcasting, e.g., BBC Big Screens. These types of deployments are characterized by closed display networks, which operate under single and central management domain offering very often, less attracting content and supporting less user interaction.

Moving beyond these isolated display systems, important research activities have been developing since 2008, aiming to open the network and attempts for building large scale public display networks. Such installations are e-Channel system deployed in Lancaster University Campus in England [3],[4] that developed and deployed an open content distribution system for display networks and UBI-Hotspot deployed in real world urban setting in Oulu from Finland [1], which focused on delivering various services through multiple interactions modalities.

Although these research initiatives and commercial display installations aimed to open the network for multiple stakeholders, none of them offers insights to support display-based application creation and distribution by third-parties.

3 Research Design

This research is organized around four key tasks, which represent the research plan and sketch the final contributions of the thesis:

1. Investigation of technologies and design space
2. Application Model
3. Evaluate open development practices
4. Evaluate scalability

Investigation of technologies and design space. This phase was primarily conceived as an exploratory research work for uncovering the design space and identifying additional research issues. In particular, it aims to identify key properties of PDNs applications, their main design alternatives and assess the viability of web technologies in supporting a set of reference applications. The task involves extensive literature review, the creation of simple applications and the study of the development environment in existing display infrastructures, more specifically the Instant Places system at University of Minho, the e-Channel system at Lancaster and the UBI-Hotspot system at Oulu.

Application model. Once the application design space is well understood, we envision that the specification of an application model for PDN should define all the necessary protocols and guidelines for allowing anyone to create this type of applications. This task should specify an application model for PDNs and evaluate systems support for display-based applications. An Agile Methodology will be followed in which a number of relevant applications will be identified for immediate implementation and technical validation of the proposed protocols.

Evaluate open development practices. Having defined a set of development guidelines, any third-party developers will be able to create and distribute applications over the network. This task is going to validate the concept of open content creation by third-parties by promoting a community of developers. Even if this is a small community it will be composed by people who are not involved in the specifications and are mainly motivated to produce working applications. The study will aim to capture the emergence of additional requirements and study the specific development practices emerging from the community.

Evaluate Scalability. Finally, the research aims to quantify the impact of performance and scalability on the proposed application model and in particular in the observed user experience. It will aim to quantify the ability of the applications to scale in the number of client display nodes and the number of interactions per client.

4 Conclusion

So far, the research activity explored the limits of web technologies for supporting the requirements for PDNs. It was investigated different application execution environments continued by first experiments with the application models for display networks.

This paper reports the work being conducted for addressing the challenges raised by the need to support third-party application development and distribution in PDN. As an outcome, the work will conceive and evaluate a web-based framework for the development, distribution and use of display-centred applications in PDN.

Acknowledgments. This PhD proposal has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 244011. Our sincere thanks also to Helder Pinto and Bruno Silva for supporting us with technical details and to the Ubicomp team involved in the PD-NET project.

References

1. Ojala, T., et al.: UBI-Hotspot 1.0: Large-Scale Long-Term Deployment of Interactive Public Displays in a City Center. In: Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services, Washington, DC, USA, pp. 285–294 (2010)
2. PD-NET Project Homepage, `http://pd-net.org/`
3. Storz, O., Friday, A., Davies, N.: Supporting content scheduling on situated public displays. Computers & Graphics 30(5), 681–691 (2006)
4. Schmidt, A., Davies, N.: Challenges in Developing an App Store for Public Displays – A Position Paper. Human Factors, 29–30 (2010)

ODBASE 2011 PC Co-chairs' Message

We are happy to present the papers of the 10th International Conference on Ontologies DataBases, and Applications of Semantics, ODBASE 2011, held in Heraklion, Crete (Greece), in October 2011. The ODBASE conference series provides a forum for research on the use of ontologies and data semantics in novel applications, and continues to draw a highly diverse body of researchers and practitioners by being part of the federated conferences event "On the Move to Meaningful Internet Systems (OnTheMove)" that co-locates three conferences: ODBASE, DOA-SVI (International Symposium on Secure Virtual Infrastructures), and CoopIS (International Conference on Cooperative Information Systems), which was also done in 2011.

We received 29 paper submissions which were subjected to rigorous reviews and discussions among the PC members. We eventually accepted nine submissions as full papers, for an acceptance rate of 31%, and an additional four submissions as short papers and five papers as posters in the workshop proceedings.

ODBASE 2011 continued its traditional focus on work which bridges common boundaries between disciplines. We believe that this emphasis on interdisciplinarity is highly valuable for the community, and provides necessary stimulus for the future development of Semantic Web research and practice. We enjoyed many fruitful discussions at the conference and a pleasant stay in Crete.

August 2011

Manfred Hauswirth
Pascal Hitzler
Mukesh Mohania

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, p. 26, 2011.
© Springer-Verlag Berlin Heidelberg 2011

FTMOntology: An Ontology to Fill the Semantic Gap between Music, Mood, Personality, and Human Physiology

Caio Miguel Marques[1], João Von Zuben[1], and Ivan Rizzo Guilherme[2]

[1] Caiena R&D, Av. 34, 13500-230, Rio Claro, Brazil
{marques,jzuben}@caiena.net
[2] UNESP – São Paulo State University, IGCE, DEMAC, Rio Claro, Brazil
ivan@rc.unesp.br

Abstract. This paper presents a domain ontology, the FeelingTheMusic Ontology – FTMOntology. FTMOntology is designed to represent the complex domain of music and how it relates to other domains like mood, personality and physiology. This includes representing the main concepts and relations of music domain with each of the above-mentioned domains. The concepts and relations between music, mood, personality and physiology. The main contribution of this work is to model and relate these different domains in a consistent ontology.

Keywords: Ontology, music semantic model, semantic retrieval.

1 Introduction

Music is part of our day-to-day life and can express emotions and, thereby, directly affect the human mind and body. In nowadays people stored their personal music dataset in electronic devices and organize their song files in playlists that represent aspects, aside from traditional music editorial information. The fact is that people organize their music datasets considering different music-related information spaces like mood, instruments, situations, personality, personal tastes, which directly impact the task of finding and retrieving tracks. Therefore a relevant problem and so some works are trying to fill this semantic gap between common music aspects and other information spaces. In this sense, ontologies could be an engineering solution to this problem. This paper presents the Feeling The Music Ontology – FTMOntology to semantically represent the complex domain of music and how it relates to other domains like mood, personality and physiology. The concepts and relations between music, mood, personality and physiology are based on our bibliographic revision and in our analysis of music datasets widely adopted in music retrieval experiments. The FTMOntology is developed in Web Ontology Language – OWL [1]. The main contribution of this work is to model and relate these different domains in a consistent ontology.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 27–28, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 The FTMOntology

In the FTMOntology was considered four main top-level concepts: Music, Mood, Personality, Physiology. The taxonomy of FTMOntology has 125 concepts, 39 properties and 359 individuals.

The *Music domain* is defined to represents general information related to music, is represented in the FTMOntology by three classes: MusicResource, MusicHighLevel-Concepts and MusicLowLevelFeatures. The MusicResource represents an music audio file. MusicHighLevelConcepts is the main class that subsumes classes such as: MusicGenres, MusicEditorialInformation, MusicInstruments, MusicAcousticFeatures, etc. The MusicLowLevelFeatures class subsumes the following classes: MusicPitch, MusicLoudness, MusicTimbre, MusicBeat, MusicFileSize, Music-SampleRate, etc. The MusicResource class relates with other classes through properties such as: hasMusicGenre, hasMusicFormat, hasMusicArtist, hasMusicBeat, hasMusicFormat, etc.

The *Mood domain* represents emotions related in a three-dimension model, represented by Valence, Arousal and Potency. In the FTMOntology is definded by ten classes: MoodHappy, MoodSad, MoodPleased, MoodBored, MoodRelaxed, Mood-Depressed, MoodAnxious, MoodAngry, MoodPeaceful and MoodExcited. These mood classes relate with MoodGenreRelation class by the property hasMoodGenre-Relation. The class MoodGenreRelation has some properties that suggest the ternary relation Mood-MusicGenre-MoodDimension.

The Personality domain is defined to represents the human personality traits based upon concepts: openness, conscientiousness, extroversion, agree-ableness and neuroticism. In the FTMOntology is defined by five classes: PersonalityExtroversion, PersonalityAgreeableness, PersonalityConscientiousness, Personality-Openness and PersonalityNeuroticism. The properties that allow the MusicGenre-Personality relation are hasPositiveRelation and hasNegativeRelation. In the Personality domain, may relate each genre of music in main characteristics of personality (e.g., Classical and Extroversion). Knowing that music is related with emotion, we may consider the level of arousal (MoodDimensionArousal) that a individual with low enthusiasm, emotionally stable, reserved and moderately open to new experiences would like to transit to feel comfortable. Thereby, even predict what kind of experiences he would be involved in pursuing its well-being (class MoodDimensionValence).

The *Physiology domain* is composed of concepts related to the main physiological indicators: heart rate, breathing rate and blood pressure. In FTMOntology we consider the Physiology class that contains three subclasses: PhysiologyHeartRate, PhysiologyBreathRate and Physiology-BloodPressure.These concepts are related to musical attributes subconcepts, like beat (MusicBeat), by four properties: canIncrease, isIncreasedBy, canDecrease and isDecreasedBy.

Reference

1. OWL. Web Ontology Language, `http://www.w3.org/2004/OWL/`

Semantics-Preserving Mappings between XML Schemas in P2P Data Integration Systems

Tadeusz Pankowski

Institute of Control and Information Engineering,
Poznań University of Technology, Poland
tadeusz.pankowski@put.poznan.pl

We discuss the problem of deciding whether there exists a semantics-preserving mapping between two XML schemas or parts of schemas specified by tree patterns. To define semantics of XML data we interpret its schema in an OWL ontology through a semantic annotation (see eg. [1]) and use this annotation to derive complex XML-to-OWL dependencies. There are two steps in our algorithm: (1) *schema matching* – identification of such maximal parts π_1 and π_2 of XML schemas, respectively S_1 and S_2, that π_1 *is semantically subsumed* by π_2 (i.e. $\pi_1 \sqsubseteq \pi_2$); (2) *schema mapping* – generating a specification describing how data structured according to π_1 are to be transformed to data satisfying π_2. The existence of matching between subschemas π_1 and π_2 is a necessary condition for the existence of semantics-preserving mapping between them.

The ontology language $\mathcal{L}$ for defining our OWL ontology is defined by its *terminology* consisting of individuals (a), text values (c), classes (C), object properties (R), and data properties (D), and rules for creating well-defined formulas (*assertion formulas and axioms*). *Assertion formulas* are: class assertions, $\texttt{CA}(C, a)$; object property assertions, $\texttt{OPA}(R, a_1, a_2)$; and data property assertions, $\texttt{DPA}(D, a, c)$ [3].

Axioms are statements in $\mathcal{L}$ expressing knowledge about individuals in the application domain. Axioms are referred to as OWL-to-OWL dependencies. We distinguish the following three kinds of axioms: (1) Axioms in the form of *tuple-generating dependencies* (TGDs): $\forall \mathbf{x}, \mathbf{v}\ (\psi_1(\mathbf{x}, \mathbf{v}) \Rightarrow \exists \mathbf{y}\ \psi_2(\mathbf{x}, \mathbf{y}, \mathbf{v}))$, where $\mathbf{x}$ and $\mathbf{y}$ are vectors of individual variables and $\mathbf{v}$ is a vector of text variables; ψ_1 and ψ_2 are conjunctions of assertion formulas. Such formulas are used to define: domains and ranges of properties, subclass/superclass relationships, existence of object- and data properties for classes. (2) Axioms in the form of *equality-generating dependencies* (EGDs) (to express functional dependencies) $\forall \mathbf{x}, \mathbf{v}\ (\psi(\mathbf{x}, \mathbf{v}) \Rightarrow x_i = x_j)$, or $\forall \mathbf{x}, \mathbf{v}\ (\psi(\mathbf{x}, \mathbf{v}) \Rightarrow v_i = v_j)$, where x_i and x_j are variables among those in $\mathbf{x}$, and v_i and v_j are amomg the variables in $\mathbf{v}$. (3) Axioms in the form of *inequality-generating dependencies* (IGDs) (used to assert class disjointness) $\forall \mathbf{x}, \mathbf{v}\ (\psi(\mathbf{x}, \mathbf{v}) \Rightarrow x_i \neq x_j)$, where x_i and x_j are variables among those in $\mathbf{x}$. An ontology over $\mathcal{L}$ is determined by the terminology $\mathcal{T}$ of $\mathcal{L}$, and a set of axioms Δ over $\mathcal{L}$, and is denoted as $\mathcal{O}_{\mathcal{L}} = (\mathcal{T}, \Delta)$.

In Figure 1 we present the idea of using shared ontology to establish correspondences (matches) between parts of schemas determined by tree patterns π_1 and π_2. Assume that a new schema appears in the data integration system (e.g.,

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 29–30, 2011.
© Springer-Verlag Berlin Heidelberg 2011

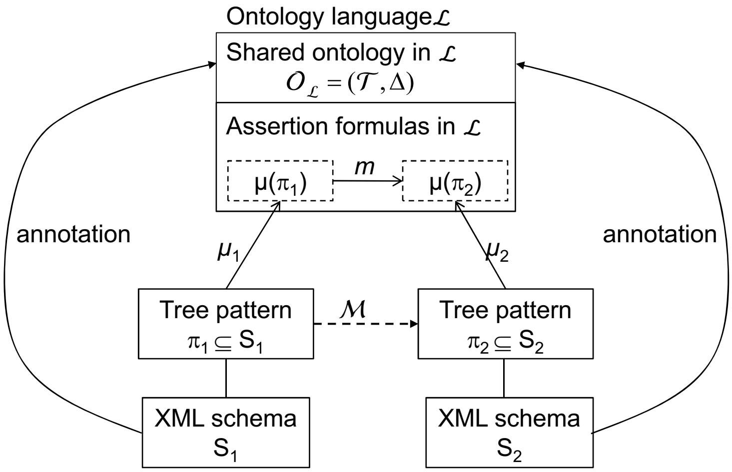


Fig. 1. Architecture of a system for discovering semantics-preserving schema mappings

a new peer with its local database enters the system). A new schema is annotated in the shared ontology $\mathcal{O}_{\mathcal{L}}$, and the process of discovering semantic correspondences between the newly introduced schema and all other schemas can be started. The matching algorithm looks for tree patterns π_1 and π_2 over schemas under discussion, and decides whether the subsumption relation, $\pi_1 \sqsubseteq \pi_2$, holds. To this order:

1. Annotations are used to create sets of assertion formulas, $\mu(\pi_1)$ and $\mu(\pi_2)$ in $\mathcal{L}$, corresponding to each tree pattern under consideration; these sets determine a mapping m that is a formula built over $\mu(\pi_1)$ and $\mu(\pi_2)$. In the result we have three dependencies (universal quantifications are omitted):
   - XML-to-OWL dependency: $\mu_1 := \pi_1 \Rightarrow \mu(\pi_1)$,
   - XML-to-OWL dependency: $\mu_2 := \pi_2 \Rightarrow \mu(\pi_2)$,
   - OWL-to-OWL dependency: $m := \mu(\pi_1) \Rightarrow \exists \mathbf{x}\ \mu(\pi_2)$.
2. It is verified whether m is derivable from the set of ontology's axioms, i.e. whether $\Delta \vdash m$ holds. If yes, then there is a XML-to-XML mapping $\mathcal{M}$ from π_1 to π_2, and the mapping is defined as the composition [2] $\mathcal{M} = \mu_1 \circ m \circ \mu_2^{-1}$.

References

1. Bellatreche, L., Xuan, D.N., Pierra, G., Hondjack, D.: Contribution of ontology-based data modeling to automatic integration of electronic catalogues within engineering databases. Computers in Industry 57(8-9), 711–724 (2006)
2. Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.C.: Composing schema mappings: Second-order dependencies to the rescue. ACM Trans. Database Syst. 30(4), 994–1055 (2005)
3. Pankowski, T.: Combining OWL Ontology and Schema Annotations in Metadata Management. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS (LNAI), vol. 6678, pp. 255–262. Springer, Heidelberg (2011)

Using Information Extractors with the Neural ElectroMagnetic Ontologies

Fernando Gutierrez, Daya C. Wimalasuriya, and Dejing Dou

Computer and Information Science Department
University of Oregon
Eugene, Oregon 97403, USA
{fernando,dayacw,dou}@cs.uoregon.edu

Abstract. In this paper, we present a study to identify instances in research papers from PubMed for the ontology developed by NEMO through the use of a component-based approach for information extraction.

1 Introduction

In this paper, we present how a novel component-based approach for information extraction named OBCIE (Ontology-Based Components for Information Extraction) [2] has been used to extract instances from PubMed[1] papers for the ontology developed in the Neural ElectroMagnetic Ontologies (NEMO) [1] project.

For the study, the OBCIE platform of *two-phase classification* [2] has been selected to develop an information extractor. The two-phase classification consists of a first phase to identify sentences containing the required information, and of a second phase to identify words within sentences that carry the actual information. The version 1.4 of the ontology developed by the NEMO project was used in the study. From the ontology, 10 classes were randomly selected for information extraction. The corpus consists in 500 abstracts that are divided into 10 sets. The abstracts come from research papers related to one of the ten concepts analyzed in this study. The research papers are obtained from the web-based archive of biomedical literature, PubMed.

2 Experiments and Results

For each class a set of words is picked as features of the class. The features are selected by contrasting their presence in sentences with and without the class (lift). If the word is relevant enough, then it is selected as feature. These features are used in the classification to help identify instances of the class.

Table 1 shows the results for *internal* and *external* performance measures. The internal performance measure compares parts of the extracted string with

[1] http://www.ncbi.nlm.nih.gov/pubmed

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 31–32, 2011.
© Springer-Verlag Berlin Heidelberg 2011

parts of the key string, whereas the external performance measure compares the whole extracted string with the key string. This makes the external measure more accurate since it compares the actual extraction with the key.

Table 1. Results for the ten classes (%)

Concept	Internal			External		
	Precision	Recall	F1	Precision	Recall	F1
EEG data set	69.40	65.95	67.63	45.52	41.79	43.57
MEG data set	80.00	62.33	70.07	66.10	57.35	61.41
Anatomical region	80.39	47.67	59.85	78.57	40.24	53.22
Anatomical surface	0.0	0.0	0.0	0.0	0.0	0.0
Brodmann area	80.24	58.03	67.35	71.79	50.00	58.94
Electrical conductivity	65.62	42.00	51.21	50.00	27.08	35.13
Electromagnetic field	74.00	58.73	65.48	68.88	60.78	64.58
ERP topography	27.54	79.31	40.88	32.51	81.53	46.49
Multimodal part of cortical surface region	75.80	62.66	68.61	79.62	58.90	67.71
Unimodal part of cortical surface region	64.00	62.99	63.49	45.45	42.01	43.66
All (except Anatomical surface)	68.56	59.97	61.62	59.83	51.08	52.75

3 Discussion and Conclusion

With an average F1 measure of 52.75%, the results of the present work can be considered acceptable when compared with OBCIE's previous study [2] that recorded F1 measures slightly below 40%, while information extraction systems such as Kylin [3] recorded F1 measures in the range of 40% - 50%. The results also show that depending on how the selection of features is done the performance of the information extraction may vary.

The application of the information extractors of OBCIE on the NEMO ontology has been successfully completed as evidenced by the acceptable performance measures obtained. Making use of such techniques in the development and enrichment of biomedical ontologies becomes increasingly important as the size and scope of such ontologies increase rapidly.

References

1. Dou, D., Frishkoff, G., Rong, J., Frank, R., Malony, A., Tucker, D.: Development of NeuroElectroMagnetic Ontologies (NEMO): A Framework for Mining Brainwave Ontologies. In: Proceedings of the 13th ACM International Conference on Knowledge Discovery and Data Mining (KDD), pp. 270–279 (2007)
2. Wimalasuriya, D.C., Dou, D.: Components for information extraction: ontology-based information extractors and generic platforms. In: CIKM, pp. 9–18 (2010)
3. Wu, F., Weld, D.S.: Autonomously semantifying wikipedia. In: CIKM, pp. 41–50. ACM, NY (2007)

SemTrace: Semantic Requirements Tracing Using Explicit Requirement Knowledge

Thomas Moser and Stefan Biffl

Christian Doppler Laboratory SE-Flex
Vienna University of Technology
Favoritenstrasse 9-11/188, 1040 Vienna, Austria
+43 (1) 58801 188051
{thomas.moser,stefan.biffl}@tuwien.ac.at

Abstract. In the software engineering (SE) domain the EU challenge on semantic systems and services motivates better support of knowledge creation as well as better access of SE stakeholders to the knowledge they need to perform their activities. Application Lifecycle Management (ALM) is the coordination of development lifecycle activities by process automation, management of relationships between artifacts, and reporting on development progress. ALM focuses on the integration of knowledge between the tools that support SE roles and thus seems particularly well suited to leverage benefits from advanced semantic technologies and services that overcome limitations from semantic gaps in today's heterogeneous SE platforms. In this paper, we present a semantic requirements tracing approach (SemTrace), which makes the implicit interdependencies between requirements and other artifacts explicit. We evaluate the proposed approach in a change impact analysis use case. Initial results show that the SemTrace approach allows for a flexibly identification of semantically connected artifacts in SE projects.

Keywords: Requirement Tracing, Requirement Knowledge, Soft-Links.

The effective and efficient creation of high-quality software products is a major goal of software development organizations where teams coming from business and technical domains collaborate to produce knowledge and artifacts. A major challenge in distributed software development is the integration of the knowledge scattered over processes, tools, and people to facilitate effective and efficient coordination of activities along the development lifecycle. Application Lifecycle Management (ALM) is defined as "*the coordination of development life-cycle activities, including requirements, modeling, development, build, and testing*".

Software engineering organizations can radically improve their project effectiveness and efficiency if artifacts can be exchanged and understood both by the human roles involved and machines, similar to advanced semantic technologies and services that facilitate the creation and usage of machine-readable semantic annotation of information. The EU research FP7 motivates a key strategic research challenge[1] in the area of semantic systems/technology development to

[1] http://cordis.europa.eu/fp7/ict/content-knowledge/fp7_en.html

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 33–34, 2011.
© Springer-Verlag Berlin Heidelberg 2011

deal with the "*growing load and diversity of information and content*" which particularly applies to SE projects where semantic support for data analysis and reasoning would improve data and knowledge exchange among distributed project teams and promises more effective collaboration.

Current ALM approaches aim at solving the technical challenges of accessing data in different tools to provide a common (relational) data model for process automation, tracing of relationships between artifacts, reporting and progress analysis beyond the data available in a single tool. ALM has been successful in the context of semantically homogeneous platforms, e.g., IBM Jazz[2] or Collabnet[3], but faces major obstacles due to semantic gaps in the integration and evolution of heterogeneous platforms, tool sets, and organizational contexts. Typical semantic gaps are multitudes of proprietary tools with their own proprietary notations, data that is scattered over multiple repositories (e.g., SVN, Mailing Lists, Bug Tracking Systems), and the fact that often just syntactic matching is used for detecting equalities between artifacts. These semantic gaps cannot be overcome efficiently with the limited syntactic approaches of current ALM approaches.

Semantic tracing builds on the semantic connections in the ALM data repository for more effective, efficient, and robust tracing of artifacts along the life cycle (e.g., to find all artifacts that help implement or test a certain requirement for change impact analysis) than traditional ALM approaches that do not use semantic knowledge. Effective tracing is demanded by SE standards (e.g., SEI's CMM-I) and a key enabler for management, design, and quality assurance activities that have to understand the relationships between artifacts along the life cycle, but currently not sufficiently well supported. Research target is to use the ontology as a means for semantic tracing in order to derive dependencies between work products (requirements, test cases, source code) from knowledge and explicit user input.

In this work we present a semantic requirements tracing approach (SemTrace), which makes the implicit interdependencies between requirements and other artifacts explicit. In the discussed use case, a change impact analysis is done for a changing requirement to find out which issues and developers are affected by the change request. This information can be used to mark all dependent artifacts for review and to contact all involved developers automatically. Furthermore it allows better estimates for the costs of the changes. Major results of the evaluation are that the change impact analysis could be defined without prior knowledge regarding the relations between the used engineering concepts, besides the fact that such a relation exists. Furthermore, the possibility to define and use semantic meta information to model relations between engineering concepts is a very flexible mechanism, which can also be used for other advanced applications, like for example automatically linking commits to related issues.

[2] `http://www.jazz.net` (IBM Jazz collaboration Platform)

[3] `http://www.collabnet.net` (Collabnet ALM platform)

Embedding User's Requirements in Data Warehouse Repositories

Selma Khouri[1,2], Ladjel Bellatreche[2], and Patrick Marcel[3]

[1] National High School for Computer Science (ESI), Algiers, Algeria
s_khouri@esi.dz
[2] LISI/ENSMA – Poitiers University, Futuroscope, France
bellatreche@ensma.fr
[3] LI, François-Rabelais University, 41000 Blois, France
patrick.marcel@univ-tours.fr

Abstract. User requirements collection and analysis have a important significance in the design of databases and data warehouses (DW). In the current situation, we notice that once the DW is designed, no tracking of requirements is kept in its model. In this paper, we propose to make user requirements model *persistent* within the DW repository.

1 Introduction

Many design methods have been proposed generating the $\mathcal{DW}$ model from (i) logical schemas of sources, (ii) from conceptual schemas, (iii) from ontological schemas. Domain ontologies may be considered as conceptual models describing an entire domain with reasoning capabilities. The definition of the $\mathcal{DW}$ model is based on two essential components: data sources and users requirements. Early design methods followed generally a data-driven approach. Many studies identified limitations of this approach and proposed requirements-driven or mixed design methods [2]. We notice furthermore that users' requirements are not used only during design phases, but are used during different exploitation phases such as optimization, personalization, recommendation and DW quality. However, no trace of these requirements is kept in the $\mathcal{DW}$ model. We propose an ontological design method, completing our previous method [3], with the main objective of ensuring the persistence of requirements in the $\mathcal{DW}$ structure.

2 Proposed Design Method

Our design method is based on two main entries: (1) OWL domain ontology (DO) integrating a set of sources, and (2) a set of user's requirements expressed as goals. Fig. 1 presents the requirements model (represented as a UML class diagram) we proposed. This model is composed of a class *Requirement* generalizing two sub classes: *user-goals* class, representing requirements gathered at the early stages of design, where we opted for a goal-oriented representation, and *user-queries* class representing the different OLAP queries, operations and analysis

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 35–36, 2011.
© Springer-Verlag Berlin Heidelberg 2011

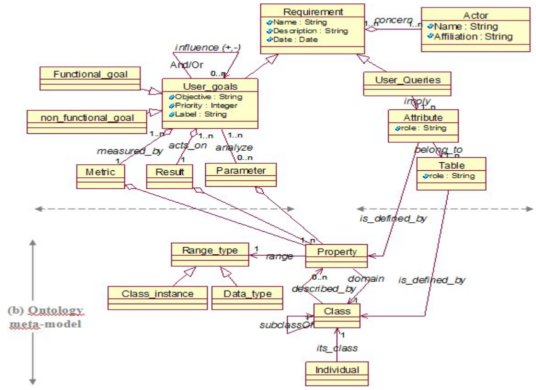


Fig. 1. Requirements model proposed

treatments, that are expressed during the exploitation of the $\mathcal{DW}$. A local ontology (LO), considered as the $\mathcal{DW}$ conceptual model, is then defined as a module of DO by extracting concepts and properties used to express goals. LO can be locally enriched with new concepts and properties. Once the LO is defined we exploit its reasoning capabilities to correct inconsistencies, and to reason about the goals model. The multidimensional role of concepts and properties are then discovered and stored as ontological annotations. The logical model of the $\mathcal{DW}$ is generated by translating the LO to a relational model. We chose an ontology-based architecture in order to store in addition, the LO defining the meaning of data. To validate our proposal, we first implemented a case tool implemented in Java language, that can be used by designers to generate the conceptual and logical $\mathcal{DW}$ model following the described steps. We then implemented the $\mathcal{DW}$ using the OBDB *OntoDB* [1].

3 Conclusion

This paper proposes an ontological method for designing a $\mathcal{DW}$, with a structure that gives a central place to requirements and represents them persistently in the $\mathcal{DW}$. This work led to some open issues: the validation and deployment of our proposal on industrial case studies and the definition of new logical $\mathcal{DW}$ models, other than the relational model.

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. Golfarelli, M., Rizzi, S.: Data Warehouse Design: Modern Principles and Methodologies. McGraw Hill (2009)
3. Khouri, S., Bellatreche, L.: A methodology and tool for conceptual designing a data warehouse from ontology-based sources. In: DOLAP 2010, pp. 19–24 (2010)

EI2N+NSF ICE 2011 PC Co-chairs' Message

After the successful fifth edition in 2010, the sixth edition of the Enterprise Integration, Interoperability and Networking Workshop (EI2N'2011) was organized as part of the OTM 2011 Federated Conferences and supported by the IFAC Technical Committee 5.3 "Enterprise Integration and Networking," the IFIP TC 8 WG 8.1 "Design and Evaluation of Information Systems," the SIG INTEROP Grande-Rgion on "Enterprise Systems Interoperability" and the French CNRS National Research Group GDR MACS. This year, the workshop was held with the First International NSF Workshop on Information Centric Engineering (ICE) and Cybertechnologies.

Collaboration is necessary for enterprises to prosper in the current extreme dynamic and heterogeneous business environment. Enterprise integration, interoperability and networking are the major disciplines that have studied how to make companies collaborate and communicate in the most effective way. These disciplines are well-established and are supported by international conferences, initiatives, groups, task forces and governmental projects all over the world where different domains of knowledge have been considered from different points of views and a variety of objectives (e.g., technological or managerial).

The past decade of enterprise integration research and industrial implementation has seen the emergence of important new areas, such as research into interoperability and networking, which involve breaking down organizational barriers to improve synergy within the enterprise and among enterprises. The ambition to achieve dynamic, efficient and effective cooperation of enterprises within networks of companies, or in an entire industry sector, requires the improvement of existing, or the development of new, theories and technologies. Enterprise modelling, architecture, and semantic techniques are the pillars supporting the achievement of enterprise integration and interoperability. The Internet of things and cloud computing now present new opportunities to realize inter-enterprise and intra-enterprise integration. For these reasons, the workshop's objective is to foster discussions among representatives of these neighboring disciplines and to discover new research paths within the enterprise integration community.

After peer reviews, 9 papers were accepted out of 20 submissions to this workshop. Paper presentation by leading experts on ICE-related topics as well as group discussions which will identify a roadmap for the future were an important and unique aspect of this year's joint EI2N and NSF ICE workshop. Niky Riga, a Network Scientist at Raytheon BBN Technologies and GENI (NSF Global Environment for Network Innovations) Project Office, was invited as EI2N+NSF ICE plenary keynote on "GENI—Global Environment for Network Innovations." She introduced GENI through a couple of example use-cases; she reviewed the growing suite of infrastructure and evolving control frameworks. She also presented previous and current experiments running in GENI. In addition to the presentations of the accepted papers, groups were organized into what E2IN traditionally calls "workshop cafés," to discuss and debate the presented topics.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 37–38, 2011.
© Springer-Verlag Berlin Heidelberg 2011

This year discussions focused around "information-centric engineering and interoperability issues" as it applies to emerging engineering and other domains. The outcomes of these respective discussions were reported during a plenary session jointly organized with the CoopIS 2011 and the OTM Industry Case Studies Program, in order to share the vision for future research with other conference attendees. The papers published in this volume of proceedings present samples of current research in the enterprise modelling, systems interoperability, services management, cloud integration and, more globally, systems engineering and enterprise architecture domains. Some new architecting principles that have gained currency in the recent past include semantic techniques and frameworks, service-oriented architectures, virtual engineering and cloud computing with their principles, reference models and technology; such frameworks and principles hold the potential to be important contributors to the future of interoperable, networked and collaborative enterprises. It was a great pleasure to work with the members of the international Program Committee who dedicated their valuable time to reviewing the submitted papers; we are indebted to all of them.

We would like also to thank all authors for their contribution to the workshop objectives and discussions.

August 2011

Hervé Panetto
J. Cecil
Qing Li

Towards Semantic Interoperability Service Utilities

Milan Zdravković and Miroslav Trajanović

Laboratory for Intelligent Production Systems, Faculty of Mechanical Engineering in Niš, University of Niš, Serbia
{milan.zdravkovic,traja}@masfak.ni.ac.rs

Abstract. The paradigm of Interoperability Service Utility (ISU) is considered as one of the grand challenges in research of Enterprise Interoperability. In this paper, we discuss on the relevance and impact of the current ISU research to the semantic interoperability in general and propose an original architecture for implementation of the Semantic Interoperability Service Utilities (S-ISU). The proposed architecture is formalized by the S-ISU Ontology – a meta-model for the semantic interoperability of the Enterprise Information Systems. Its intended purpose is to be used as abstract model for design of the organizational and asset perspective of the systems interoperability in a virtual enterprise.

Keywords: Semantic Interoperability, Ontology, Virtual Enterprise.

1 Introduction

The notion of interoperability refers to the ability of heterogeneous, autonomous enterprise information systems (EIS) to perform interactions (exchange of information and services) [1]. It is related to the federated approach, which implies that systems must accommodate on the fly in order to interoperate – no pre-determined assets are assumed. While the notion of enterprise integration is usually associated to the extended enterprises, interoperability is exploited in the virtual enterprises - temporary networks of independent enterprises.

Interoperability is expected to become one of the basic IT functionalities for business operations which enterprises can exploit in the form of Interoperability Service Utility (ISU) services [2]. ISU is envisaged as a transparent, scalable, vendor-independent infrastructure (system), built upon the paradigm of SaaS (Software-as-a-Service), leveraging open standards, available and accessible to all and by all. Here, we discuss on some of the reported work in this field and in specific – about the role of ontologies for achieving the enterprise interoperability.

There is an agreement in the research community that ontologies need to be used for reconciliation of the interoperating systems. Even so, there are opinions that the main conditions for interoperability are to maximize the amount of semantics which can be utilized and to make it explicit [3]. In this sense, we believe that it is important to distinguish between two approaches to interoperability of systems. The first is using ontologies as facilitators for interoperability - mediators of individual business concepts, such as documents, software applications, etc. The second approach assumes that interoperable world is made of ontologies. It consists of EISs' implicit or

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 39–48, 2011.
© Springer-Verlag Berlin Heidelberg 2011

explicit formal models, mapped to the formalized standard models, which are then used as dictionaries to enable universal understanding of the interoperable enterprises. Hence, we distinguish between ontology-facilitated and semantic interoperability.

In this paper, we address the problem of ISU for semantic interoperability of EISs. In Section 2, we discuss on different interoperability issues and ISU research directions. Then, we identify the Semantic Interoperability Service Utilities (S-ISU), make some design decisions and describe the overall architecture for their implementation in the Section 3, formalized by using proposed meta-model, namely S-ISU ontology.

2 Interoperability, Semantic Interoperability and ISU

Enterprise interoperability may be considered and evaluated on multiple levels, such as: information/data, services, processes, systems, enterprise models, enterprises and communities. Each of the levels is characterized by the specific challenges. For example, data formats without semantics are the main issue of information interoperability; static definition is serious restriction for services interoperability; lack of correspondences between standard and implemented models and realities poses the challenge for enterprise models interoperability. Although each of these challenges can be associated to a particular level of interoperability, they cannot be addressed in isolation. Namely, EISs capture implicit knowledge of the enterprise; systems are exposed by their services, which are then used to exchange information through enterprise or cross-enterprise processes. Thus, only holistic approach to enterprise interoperability can produce the knowledge and associated assets for realizing its value proposition. The ATHENA Interoperability Framework [4] adopts such a holistic perspective by inter-relating three research areas supporting the interoperability of EISs. The three areas are: 1) enterprise modeling (which defines interoperability requirements), 2) architectures and platforms (which provide implementation frameworks), and 3) ontology to identify interoperability semantics in the enterprise. ATHENA identifies the levels where interoperations can take place: enterprise/business, process, service and information/data. Then, for each of these levels a model-driven interoperability approach is prescribed, where meta-models are used to formalize and exchange the provided and required artifacts that must be agreed upon. In ATHENA Framework, the semantic tools, namely, ontologies are intended to be used as facilitators for the interoperability. We believe that it's important to distinguish semantically supported interoperability from the semantic interoperability as the latter goes beyond mere data exchange and deals with its interpretation.

Semantic interoperability of systems means that the precise meaning of exchanged information is uniquely interpreted by any system not initially developed for this purpose. It enables systems to combine and consequently process received information with other information resources and thus, to improve the expressivity of the underlying ontologies. In our research, we adopt the formal definition of John Sowa [5]: "A sender's system S is semantically operable with a receiver's system R if and only if the following condition holds for any data p that is transmitted from S to R: For every statement q that is implied by p on the system S, there is a statement q' on the system R that: (1) is implied by p on the system R, and (2) is logically equivalent to q.”. Here, system S and R are represented by the so-called local ontologies.

In our work, we assume the following: when two local ontologies of two corresponding systems are mapped to the same domain ontology, these systems will become semantically interoperable. In other words, if there exist two isolated EISs S and R and corresponding local ontologies O_S and O_R and if there are mappings M_{SD1} and M_{RD1}, established between the concepts of O_S, O_R and domain ontology O_{D1}, respectively, then there exist mappings M_{SR} which can be inferred as logical functions of M_{SD1} and M_{RD1}. Local ontologies are considered as the models of implicit enterprise knowledge. This knowledge is made explicit and hence, machine-processable, when implicit terms of the local ontologies are logically related to appropriate enterprise conceptualizations (e.g. standard models), represented by domain ontologies. Furthermore, each of the local ontologies may represent one of the contexts of the enterprise. Hence, the isolated systems become not only interoperable, but also more expressive, as they become capable to exploit enterprise knowledge, represented by the different local ontologies. Expressivity can be improved further when focal domain ontology is related to another domain ontology in the same manner. This approach may be exploited for the benefit of assertion of the enterprise knowledge by using different conceptualizations, encoded in the different domain ontologies.

When considering the above principles of the semantic interoperability of systems, it can be concluded that it is unconditional and universal. It is not structured by the levels, nor does it assume the particular kind of architecture for its implementation (by using, for example, SaaS paradigm). Restrictions may occur, but they can be only related to: a) incompleteness and lack of validness of logical correspondences between two ontologies; b) expressiveness of the implicit models, namely local ontologies; c) expressiveness of the languages, used to formalize those models; or d) restricted access to some of the information, modeled by the parts of local ontology.

In this section, we refer to the work of Zhang et al, iSURF and COIN FP7 projects in ISU development and briefly discuss on their main features, in context of what we need to develop the S-ISU architecture. Zhang et al [6] introduced the ISU for automobile Supply Chain Management which enables establishing interoperability between two SaaS applications, by: (1) constructing a virtual enterprise; (2) dynamically composing available ISUs according to requirements; (3) evaluating and improving the interoperability solution. The framework delivers interoperability as utility-like capabilities in the form of SaaS. The main functionality of iSURF ISU [7] is to perform the semantic mediation of planning documents across enterprises by using a common denominator, OASIS UBL documents. Universal Business Language (UBL) is a framework consisting of library of standard electronic XML business documents, such as purchase orders and invoices and customization methods. iSURF ISU platform is extended to provide interoperability services to all CCTS (UN/CEFACT Core Components Technical Specification) standards based documents, such as UBL, GS1 XML and OAGIS, enabling the inference of the relationships between different artifacts of different standards. Where previous two cited works are development-oriented, Elvesæter et al [8] identify Enterprise Interoperability Services on the basis of the ATHENA Interoperability Framework. The approach builds upon ATHENA framework's dimension, to classify interoperability services into 1) Model-driven, 2) Enterprise modeling, 3) Business process, 4) Service, 5) Semantic mediation and 6) Information and data interoperability domains.

One of the design principles of ISU is that it exploits services. Thus, interoperability becomes intentionally restricted and partial because it depends on their scope and functionality. The restrictions of service-oriented approach can be considered in two aspects. First is related to the scope and availability of existing enterprise services which is a precondition for ISU implementation. Second aspect of the conditionality is related to variety and diversity of interoperability services. For example, in the work of Zhang et al, ISU services layer is organized in a functional way, where SBM (Supply Business Management) and APO (Advanced Planning and Optimization) ISU services are considered as the most important. Although the interoperability restrictions are not direct implication of the functional organization approach, it is obvious that, in this case, the scope of interoperability between two systems will depend on the variety of available functional interoperability services.

The scope restriction is even more evident in the work of iSURF ISU development. iSURF platform is based on the document models. Hence, its purpose can be considered more like syntax than semantic interoperability. Semantic mediation is the only ISU service in iSURF. It reconciles the models of very low level of abstraction. This, bottom-up type of approach contradicts to a usual practice of ontology engineering, where new specific concepts are inherited from general notions residing in upper ontology, so a consistent and expressive ontological framework is built. However, bottom-up approach also has some advantages. First, it is usually built upon the implicit, but common, widely accepted knowledge (in this case, CCTS set of standards). Second, the development time is shorter, because the process of ontology engineering is reduced to semantic analysis of the documents standards. Finally, the evaluation problem can be only reduced to consistency checking and completeness assessment.

3 Semantic Interoperability Service Utilities

In this section, we identify the Semantic Interoperability Service Utilities and make certain design decisions about the conceptual architecture. Then, service utilities are described in the architectural context. This description is formalized in the S-ISU Ontology for semantic interoperability of EISs.

The focal problem of semantic interoperability of systems is identification of the logical correspondences between two models. Hence, the most important service in the S-ISU architecture is Semantic Reconciliation Service (SRS). The process of recognition and, in some cases, assertion of those relations, corresponds to the ontology operations: merging, mapping, alignment, refinement, unification, integration or inheritance. Those tasks are difficult and cannot be performed automatically in non-trivial cases. Typical reasons are usage of very expressive languages which may result with undecideability or insufficient specification of conceptualizations. Obviously, SRS must be coupled with client software, which needs to facilitate review and approval of suggested generated mapping axioms, as well as manual assertions.

In semantic interoperability architecture, an enterprise is introduced by its ontology or ontologies. We call these ontologies – local ontologies. The local ontology may be any formal model of the enterprise or any of its contexts, which describes some reality of an enterprise, and with which the enterprise wants to be represented in the interoperable world. Introduction of the enterprise is enabled by the Registration Service (RegS). It facilitates declaration of the local ontology (or ontologies) location and rules for semantic queries handling. Namely, enterprise may decide to unconditionally restrict access to specific information (sub-graph) in the local ontology. Or, enterprise may want to be capable to manage access to particular information per request in the process of query execution. It is important to note that, in latter case, the process of semantic querying will become asynchronous. RegS is also used for registering the domain ontologies. These ontologies describe different perspectives to an enterprise or one of its contexts, namely, standard dictionaries which are used to infer about the enterprises local ontologies specific, implicitly defined concepts.

The local ontology is representation of the implicit semantics of an enterprise. If we assume that the realities of an enterprise are stored in the corresponding EISs, we can identify their relational databases and other data storage facilities as valid sources of this semantics. These sources need to be exposed in a certain way, in order to enable the transformation of the implicit enterprise knowledge they contain to a valid local ontology. Thus, the Transformation Service Utility (TrS) is identified as an element of S-ISU architecture. This utility is already developed and described in detail by Zdravkovic et al [9]. The approach enables the complete (from the aspect of OWL expressivity) explicitation of the implicit semantics of the ER model, as well as full correspondence between semantic and database queries. This correspondence is exploited in the design of the Semantic Query Service.

The Semantic Query Service (SQS) is considered with "Ask" and "Tell" interfaces, enabling extraction of relevant instances and assertion of new ones in designated local ontologies. SQS is a single point of access to the overall knowledge of the interoperable world. Its "Ask" interface accepts semantic (e.g. DL – Description Logics) queries in the form of a pair (O, C), where O is a set of concepts which need to be inferred and C - a set of restrictions to be applied on their properties. When mappings between registered local and domain ontology(s) are consistent and complete, one can use the dictionary of the domain ontology(s) to build semantic queries, without any knowledge on the underlying semantics of the enterprise local ontologies [9]. The "Tell" interface of the SQS, accepts semantic queries in the form of a triple (A, C, U), where A is a set of assertion statements, C - a set of conditions represented by a common dictionary(s) concepts and U – identifier of the local ontology where assertions need to be made.

Reasoners are used when semantic queries are issued or in the process of semantic reconciliation. A first step towards the provision of reasoners that can be deployed in a distributed architecture is the Description Logics Implementation Group's specification of the DIG Interface [10]. It accepts HTTP requests and responds accordingly with the content defined by an XML schema. Since DIG is simply a protocol that exposes the reasoner, it does not support stateful connections or authorization. Hence, a Semantic Reasoning Service (ReaS) is anticipated in the architecture, to be implemented on the top of the DIG interface with provision of functionality which is not inherently supported. Almost all the work on semantic reasoning still assumes a centralized approach where all inferences are carried out on a single system. However, transfer of the complete model to a central reasoner takes time and reasoning systems have limited performance [11]. Thus, ReaS is envisaged as distributed service.

On the basis of above analysis, we propose the architecture for achieving the semantic interoperability of the EISs, namely, S-ISU architecture. It consists of the ontological and utility frameworks, located and exploited centrally or locally. Namely, some of its assets are located behind the enterprises firewalls, while others are shared by the pool of enterprises, or owned by its broker. Figure 1 shows the component view of the S-ISU interoperable world's architecture.

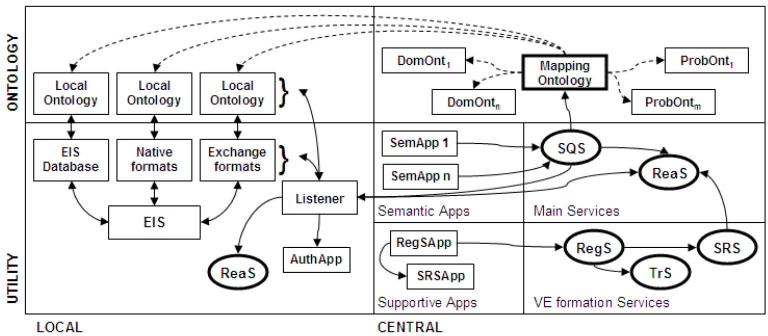


Fig. 1. Component view of the S-ISU architecture

Locally, enterprises introduce their implicit semantics, residing in the EISs' databases, native and exchange formats, etc., to the interoperable world, by using local ontologies. They are mapped then to an arbitrary number of centrally stored domain ontologies ($DomOnt_{1-n}$), which formalize the dictionaries, so one can query the local ontologies of unknown structure, by using terms from known models. At the central level, we also introduce so-called application, or problem ontologies ($ProbOnt_{1-m}$), which are used to formalize specific, integrative functions of the virtual enterprises, e.g. collaborative business process management or biding. Problem ontologies are used then by the shared semantic applications which facilitate these functions ($SemApp_{1-m}$). In Figure 1, ontologies are mutually related by import relations (dashed lines). Other relations between components are of "uses" type. In this architecture, we distinguish between the services which are used during the lifecycle of the virtual enterprise (SQS, ReaS) and those which are used only ones, in the process of its formation (RegS, SRS and TrS). As mentioned before, the supportive applications (RegSApp and SRSApp) are introduced in order to facilitate a human involvement in the processes of registration and semantic reconciliation. We consider their inner workings as trivial, so we will not discuss them in detail.

Single point of access to an interoperable world is provided by SQS, namely its "Ask" and "Tell" interfaces. They accept the semantic queries, built by a user, a semantic application or another service. Upon receive, the "Ask" query (built by using one of the registered dictionaries) is interpreted "in the languages" of each of the registered local ontologies. This translation is done by ReaS, based on the mappings between used dictionary and the local ontologies. Then, the local queries are launched concurrently. Local query execution is performed by the listeners, local components of the S-ISU architecture. They accept the individual requests for information and launch the queries. If the enterprise decides to host a dedicated reasoner, then it is used for inference of the query results. Otherwise, a central reasoning service is invoked. Based on the access rules, the results (OWL triples) enter the approval procedure (facilitated by Authorization Semantic application – AuthApp, to approve or deny requests) or are immediately delivered back to the SQS.

The above architecture is formalized by the S-ISU ontology. Its main concept is a Component, which classifies Interface, Data-Container and Utility concepts. Other top level concepts are Actor, Process, Data and Function (only used to aggregate natural language descriptions of the functions). In the context of interoperability, an Interface is the main functional component of S-ISU. Data-Container is any component which involves some kind of data persistence, asserted by “stores” relationship, and aggregates the concepts of Database, File and Ontology. Utility is an abstract concept which subtypes are Enterprise-Information-System, Listener, Semantic-Application and Service. An Actor is defined as something that uses some utility. It classifies employees, departments, enterprises, Virtual Breeding Environments (VBE) and Virtual Enterprises (VE), while additional properties describe relationships between those. These relationships may be used to infer the accessibility of a particular utility by specific actor, based on the ownership and collaboration properties. More important, the relationships can point out where interoperations between enterprises in a VE take place. Namely, we consider a VE as a set of processes, configured by simple precedence relations. Then, VE is assembled of the enterprises which implement its processes. Thus, partnership relation of the enterprise in specific VE is inferred as a property chain implements-process(Enterprise, Process) o is-process-of(Process, VE). Each of the processes is assigned to an individual enterprise in the process of VE formation, while additional assertions are made to declare which EISs owned by the enterprise facilitate the specific process. Interoperations between two enterprises occur when a process, owned by one enterprise, precedes (or succeeds) the process of another. Hence, enterprise interoperation relationships may be inferred by using SWRL rule: Process(?p1), Process(?p2), Enterprise(?e1), Enterprise(?e2), implements-process(?e1,?p1), implements-process(?e2,?p2), precedes(?p1,?p2), DifferentFrom(?e1,?e2)->interoperate-with(?e1,?e2). Key concepts and properties for making this inference are presented at Figure 2a. Figure 2b shows example processes (with asserted precedence relationships) of the VE for snow making facility engineering, assembled of three enterprises, where implements-process property is illustrated by the pattern of the enterprise and process individuals. Based on a rule above, following inferences are made: interoperate-with('Pumps-Inc', 'Snow-Solutions-Inc'), interoperate-with('Lenko-Snow-Inc', 'Snow-Solutions-Inc'), interoperate-with ('Snow-Solutions-Inc', 'Lenko-Snow-Inc') and interoperate-with ('Snow-Solutions-Inc', 'Pumps-Inc').

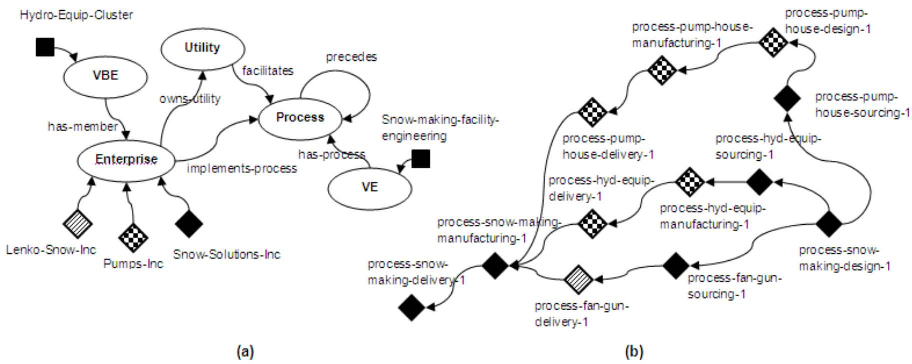


Fig. 2. Organizational view of S-ISU Ontology

While figure 2 illustrates portion of the organizational view of S-ISU ontology, component architecture is described by its asset view, generated by dependency relationships inference. Dependency analysis is generated by inferring "uses" relationships between the components of S-ISU, on basis of the asserted sub-properties of transitive 'uses' property, such as imports(Ontology, Ontology), uses-data-container(Utility, Data-Container) and uses-utility(Utility or Actor, Utility). Where latter is used to assert the interoperation relations between generic utilities in S-ISU, interoperation between registered (asserted) EISs is inferred by using SWRL rule: EIS(?u1), EIS(?u2), Process(?p1), Process(?p2), Enterprise(?e1), Enterprise(?e2), facilitates(?u1, ?p1), facilitates(?u2, ?p2), implements-process(?e1,?p1), implements-process(?e2,?p2), precedes(?p1,?p2), DifferentFrom(?e1,?e2)->system-interoperate-with(?u1,?u2). It is important to note that interoperation properties are not symmetric, because we consider semantic interoperability of systems as unidirectional. Dependency analysis is demonstrated on the example of the snow making facility manufacturing supply chain, where ontological framework for semantic interoperability, based on Supply Chain Operation Reference (SCOR) model [12] is applied. Asset view of the S-ISU architecture in this case is shown on Figure 3. The illustration distinguishes between asserted components (depicted by rhombs of different patterns, depending on the ownership of the corresponding components) and generic components of S-ISU (depicted by squares), both individuals of S-ISU ontology. Also, membership of the individuals to S-ISU concepts (Oval symbols) is asserted and shown on the figure (solid line). Asset perspective of the S-ISU architecture on figure 3 is illustrated by example supply chain, where three enterprises are interoperating in the organizational context, shown on figure 2. In this case, enterprises expose ERPNext's MySQL and OpenERP's PostgreSQL databases and EasySCOR system native format to VE for snow making facility manufacturing, by using local ontologies: ERPNext-1-Ont, OpenERP-1-Ont and SCOR-KOS OWL, respectively.

Two shared semantic applications are facilitating the VE's lifecycle, namely, SCOR-Thread-Gen, for supply chain process configuration; and Prod-Acquis-App for acquisition of product requirements, where respective problems are modeled by two application ontologies: SCOR-CFG and PRODUCT-OWL. Both applications are using SQS to assert to or infer about the implicit knowledge in local ontologies, by using two dictionaries: TOVE Enterprise Ontology and SCOR-FULL – semantic enrichment of the SCOR reference model.

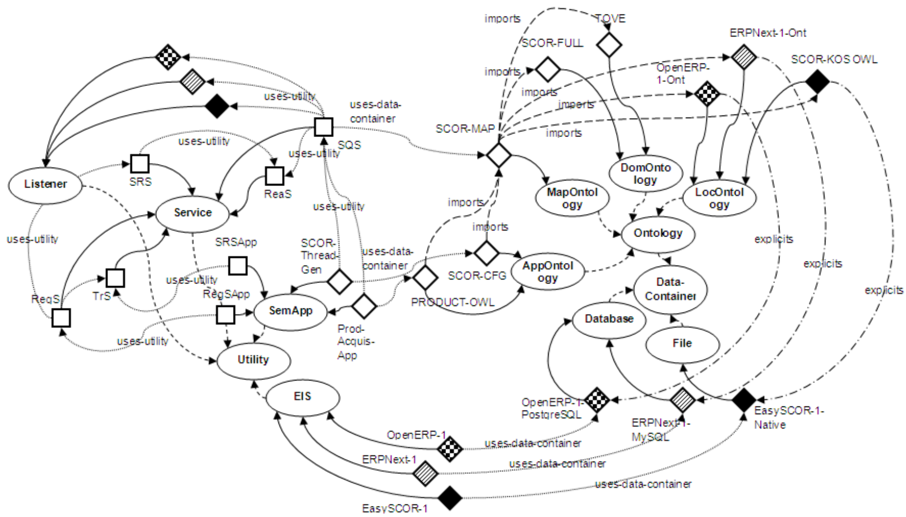


Fig. 3. Asset view of the S-ISU architecture

4 Conclusion

To some extent, presented approach revises the conceptual directions of ISU development. While ISU is based on orchestration of the services – islands of EISs' functionality, our approach assumes that enterprises are represented in the interoperable world by the islands of semantics. This semantics is interpreted by the ontological framework, where standard dictionaries are used to infer about or assert to implicit enterprise realities, by exploiting mappings between corresponding domain and local ontologies. In addition, a pool of enterprises may implement shared business functions, described by the application ontologies and facilitated by the shared semantic applications. The ontological framework is then exploited by a set of utilities which provide registration, transformation, semantic reconciliation, semantic querying and reasoning services. S-ISU ontology's intended purpose is to be used as a meta-model for design of the VE's organizational and asset view, where services above are considered as generic. It also infers about interoperation relationships in VE, utility accessibility and other. Thus, it reflects the model-driven engineering (MDE) paradigm.

The semantic technologies are still considered immature for a large-scale implementation, where lack of performance becomes a serious issue. Modular ontology frameworks and distributed reasoning may reduce its negative effects, but the challenges of all-pervasive application of the semantic tools still remain. Hence, ontology-facilitated interoperability is more likely to become a reality in a short term, than semantic interoperability. However, we believe that presented arguments clearly demonstrate benefits of the unconditional interoperability. In conventional ISU approaches, the scope of interoperability may be considered in horizontal (interoperability levels) and vertical directions (business functions). In both, restrictions may occur in terms of implementation decisions or initial requirements. In S-ISU approach, the only restriction is related to the amount of available interpretable enterprise semantics. Thus, it is universal, problem domain independent and scalable in the sense that it only depends on the expressivity of the enterprise semantics.

References

1. Chen, D., Vernadat, F.: Standards on enterprise integration and engineering—a state of the art. International Journal of Computer Integrated Manufacturing 17(3), 235–253 (2004)
2. Enterprise Interoperability Research Roadmap, `http://cordis.europa.eu/ist/ict-ent-net/ei-roadmap_en.htm`
3. Obrst, L.: Ontologies for semantically interoperable systems. In: 12th International Conference on Information and Knowledge Management. ACM, New York (2003)
4. Berre, A.J., Elvesæter, B., Figay, N., Guglielmina, C.J., Johnsen, S.G., Karlsen, D., Knothe, T., Lippe, S.: The ATHENA Interoperability Framework. In: 3rd International Conference on Interoperability for Enterprise Software and Applications, pp. 569–580. Springer, Heidelberg (2007)
5. SUO, The IEEE Standard Upper Ontology web site (2001), `http://suo.ieee.org`
6. Zhang, Y., Lio, S., Jiao, Y.: An interoperability service utility platform for automobile supply chain management. In: 12th Enterprise Distributed Object Computing Conference Workshops, Munich, pp. 230–237 (2008)
7. Dogac, A., Laleci, G.B., Okcan, A., Olduz, M., Sesana, M., Canepa, A.: iSURF: An Interoperability Service Utility for Collaborative Supply Chain Planning across Multiple Domains. In: eChallanges 2008 Conference (2008)
8. Elvesæter, B., Taglino, F., Del Grosso, E., Benguria, G., Capellini, A.: Towards Enterprise Interoperability Service Utilities. In: 12th Enterprise Distributed Object Computing Conference Workshops. IEEE Computer Society, Washington, DC (2008)
9. Zdravković, M., Trajanović, M., Panetto, H.: Local ontologies for semantic interoperability in supply chain networks. In: 13th International Conference on Enterprise Information Systems. SciTePress (2011)
10. Bechhofer, S., Patel-Schneider, P.F.: DIG 2.0: The DIG Description Logic Interface Overview, `http://dig.cs.manchester.ac.uk/overview.html`
11. Schlicht, A., Stuckenschmidt, H.: Distributed Resolution for Expressive Ontology Networks. In: Polleres, A., Swift, T. (eds.) RR 2009. LNCS, vol. 5837, pp. 87–101. Springer, Heidelberg (2009)
12. Zdravković, M., Panetto, H., Trajanović, M.: Towards an approach for formalizing the supply chain operations. In: 6th International Conference on Semantic Systems. ACM, New York (2010)

Interoperability Analysis: General Concepts for an Axiomatic Approach

Michele Dassisti[1] and David Chen[2]

[1] Politecnico di Bari, D.I.M.e.G., Viale Japigia 182, 70126 Bari, Italy
m.dassisti@poliba.it
[2] IMS, University Bordeaux 1, 351 Cours de la liberation, 33405 Talence cedex, France
david.chen@ims-bordeaux.fr

Abstract. The purpose of the paper is to provide general criteria and evidences for the design phase of an interoperable enterprise system. The interoperability analysis is introduced, to characterize features and criticalities for the subsequent design actions to be undertaken. An axiomatic approach is proposed to this aim, by providing general contextualized principles to be followed. Discussion and further research are given as part of conclusions.

1 Introduction

Interoperability (Io) is the ability of one (or more) system to exchange information and use information exchanged [9], or it can be extended as the ability to use the functionality of another system [16]. From system theory point of view, interoperability can be defined, as the ability of a system to cooperate with one or more other systems to pursue a common goal within a definite interval of time [4]. In other words, interoperating for a system means to provide a mutual flow of desired outputs (information, products...) to another system so that the cooperation becomes effective. It is clear that, in order this definition being non recursive, whenever interoperating two or more systems form a unique temporary system which is expected to share a common goal. The concept of system is here used in its general meaning: it can refer to an enterprise, or any physical or logical part of it. It is also here assumed that any problem related to system interoperability requires a decision making process: no unmanned interoperation process is supposed possible so far.

Any decisional process is characterised by an observation process of the system reality by the decision maker, who creates his own abstraction of reality using a mental model [5] [3], which will be constantly used by himself in any decision making. Unavoidably, a number of details of the system observed are neglected in this mental model, due to the abstraction process made and according to the decisional process to be performed. This model is also subjective, due to the strong influence of personal experience and judgment in the modelling process, i.e. it is possible that different decision makers generate different models of the same system for the same scope.

Several methods are possible to make explicit decisional models; one of the most recent and appreciated explicit representation methods of decisional models is Enterprise Modelling (EM) (see [2], [3] and [17]). Logical constructs behind EM aims at reducing the complexity of the decisional process to be performed. Using a specific

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 49–60, 2011.
© Springer-Verlag Berlin Heidelberg 2011

EM approach means to implicitly adopt all the assumptions and hypothesis on which it relies, which are strongly related to the domain in which it is used.

On the other hand, in EM modelling for interoperability it is important to refer to general behavioural principles and modelling criteria, not so frequently explicated in many EM approaches, to effectively support decision making process.

An axiomatic approach is here proposed to this aim, to support design of enterprise interoperability independently of the specific domain. It is intended to provide a set of information and knowledge useful to support the decision-making process. Stemming from the axiomatic theory of Suh [15] or in El-Haik [6], the idea of an axiomatic approach descends from the fact that enterprises or manufacturing systems are rarely interoperable per se. A higher effort is required to embed this feature which, on the other hand, seems already well operated in natural systems (say biological organisms, groups of persons, etc.). The axiomatic approach introduces some principle of rationality derived from what is already known in the general system theory by Bertalanffy [1] and reflects the current understanding on interoperability concept. Consequently it aims to be a meta-approach, general enough to support a wide variety of enterprises and industrial sectors.

2 Axiomatic Approach for Interoperability (Io) Analysis

According to Suh [15], it is possible in principle to recognise general rules for best performing a given design activity. Accordingly, a set of axioms need to be firstly defined, in order to make each design step consistent.

The design activity may concern several goals: in general terms, the design of systems to assure their interoperability is the newest and less experienced one. Despite the well assessed set of knowledge in the I.T. domain, this task is still not clearly defined at enterprise level, where different types of systems are involved: technological, organisational, physical, etc.. With the scope to provide a structured path to the design a generic system so as to be interoperable, we selected the axiomatic approach because of the possibility to explicit concepts not yet fixed so far, and also the chance to set logical steps to build an effective mental model for decisional processes.

The very first step of any design activity is the analysis and definition of the reasoning domain: this first step is here called the "interoperability (Io) analysis" and will be the object of the present paper. Performing an Io analysis means to define objects that will be of interest of the system under analysis, as well as the scopes of the analysis. These latter will act as constraints for the subsequent decisional processes, and referring to generic systems should answer to the question "why to interoperate?". As an outcome of the Io analysis aims there is the definition of the systems (or parts of it) to be considered, the time gap they are expected to support interoperation, and the responsibilities for interoperation. This interoperability analysis turns to be the essential part of the Io design process, particularly because inter- and intra-relationships need to be appreciated amongst the domain objects involved.

Let firstly give general definitions of the terms we will use so forth by referring to figure 1, according to the system theory of Bertalanffy [1]. A system (without reference to special cases or situations) can be interpreted as made by subsystems (say, for instance, functional parts of it): a first problem is to decide the boundaries of these, i.e. their mutual exclusiveness.

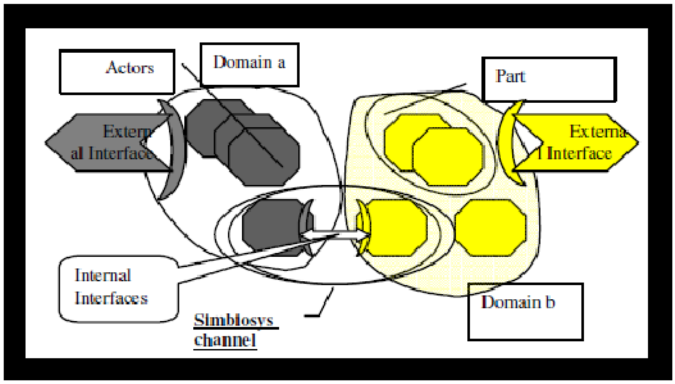


Fig. 1. Logical schema of a generic system

Subsystems are made of actors that can be physical resources or human resources. The exclusivity property of subsystems requires that actors belonging to only one subsystem. A part of a subsystem is constituted of a set of actors belonging to it. Furthermore, a sub-system may be also a set of heterogeneous actors, i.e. belonging to different subsystems. With the term *components* we will refer generically to any subsystem, parts, sub-parts or actors. Interfaces in fig.1 act as connecting elements between components; these will be defined in detail later in the paper. It is worth to note from the logical schema in fig.1 the "symbiosis channel" that is a set of elements and interfaces that provides the exchange of elements (information, parts of any kind) useful to the interoperation.

2.1 Setting the Axioms for Interoperability Analysis

Axiomatic approach is widely used in various scientific research areas. An axiom is as a general and commonly agreed statement, based on experience or sound reasoning; for scopes of the present paper these were collected from a wide bibliographical analysis in the scientific literature on similar subjects (see [14], [7], [11], [12]) and also authors' direct knowledge on the field. Descending from the set of axioms, it results easy to derive the steps for an effective interoperability analysis by using first-order logic reasoning.

Here following the four set of axioms (and sub-axioms) are given, providing sound reasoning for their assumption, assuming that an Interoperability analysis (Io analysis) has to be performed.

AA – Analysis axiom

Whenever two or more systems have to interoperate, it is mandatory to define explicitly their components, features and relationships; that means to make explicit the subjects involved (WHO), the time gap of the interoperation (WHEN), the component and resources involved (WHAT), the arrangement of components and resources to be used (WHERE) in order to allow the correct definition of the method of interoperation.

MMA – Metamodellig axiom

Therefore, *In order to derive general conclusions from an Io analysis it is important to derive an instantiated model starting from a metamodel.*

A metamodel is a general model, therefore not useful for driving any decision by its nature. Models useful for decision in a specific application are instantiated models. An instantiated model can be build from a meta-model or from scratch. An instantiated model is necessarily influenced by the decision-making scope of its use (e.g. designing, process reengineering, etc.): that means any instantiated model is a function of the scope of its use. This implies that it is not possible to build an "inertial instantiated model", i.e. independent of the specific application scope (or from the decision makers). As a consequence of this it is logical to state that an instantiated model needs to be derived from a metamodel, which is not domain specific.

MA – Modelling axiom

In order to analyse interoperability between two (or more) systems it is necessary to refer to an instantiated model of those parts of the systems involved in the interoperation and their relationships (the object of reasoning).

This needs to be made with appropriate modelling criteria, with an assigned degree of abstraction and level of granularity, in order to allow driving appropriate decisions.

MA1 – Complexity sub-axiom

In order to effectively interoperate, the abstraction level of the model needs to be accurately identified in such a way to minimize the level of complexity of the decisional process. The lower the degree of detail of the model (level of granularity or number of components included) is, the higher the complexity in decisional process and its risk of useless will be. The vice versa also holds.

MA2 – Decomposition sub-axiom

In an interoperation process, all the components (subsystems) of the systems and their inter- and intra-relationships need to be clearly identified. In order to effectively model two (or more) interoperating systems, a set of fundamental actors and their relations need to be identified within the components. These actors should be identified in a manner that it is possible to decompose and rebuild the system models (degree of abstraction) any time with the minimum loss of information.

MA3 – Specificity sub-axiom

Actors responsible for interoperation might require different features as a function of the specific subsystems. Therefore, *several fundamental actors need to be* *recognised for each subsystem of the system involved in the cooperation. In general, it is not possible to derive conclusions on interoperability without referring to subsystems.*

SA – Systemic axiom
A system is a harmonic whole of inter-related subsystems. *Subsystems need to be characterised by actors having intra-features (i.e. features concerning the actors operating inside the system) and inter-features (features concerning the ability of actors to interface with external actors).* The system behaviour to any stimulus is a function of the intra- and inter-features defined for its actors.

IA – Interfacing axiom
Interoperating requires an interaction between two or more subsystems or actors of a system. *A clear identification of "interfaces" and their features and constraints (either internal or external) is required. Whenever a transaction of any nature takes place (i.e. exchange of information, knowledge, product, etc.) through interfaces, this implies a state-transformation of their features and, consequently, a transition between states of the whole system occurs.*

IA1 – Interface feature sub-axiom
Whenever performing an Io analysis it is customary to identify interfaces, their pro-active actions to promote cooperation, whether explicitly recognised or implicitly performing as interface.

As external interface is intended a whole subsystem or a set of actors (even a single actor), belonging to the same or to different subsystems of a system, which is deputed to foster communication between different subsystems. This external interface is said to have inter-features. Also internal interfaces are present, allowing communication and cooperation between parts of the same system. This internal interface is said to have intra-features

3 Implications of the Axiomatic Approach

Given the sets of axioms provided in the section 2.1, several actions can consequently be derived to perform an interoperability analysis.

Implications of the AA axiom: recognising the components of the reasoning
The analysis axiom states the needs for an Io analysis, and traces the elements of the reasoning domain. It states that it is not possible to design interoperability without clearly recognising all the critical elements of the domain. .

Implications of the MMA axiom: setting the scope
The metamodel axiom states that it is not possible to build a general model, valid for any decisional situation. It is therefore clear that also outcomes of the interoperability analysis cannot be performed independently of the system or the design purpose for which it is intended for. This requires a constant reference to the analysis scopes as

well as (axiom MA) an upgrade or revision of the instantiated models as far as the scope or the same system changes.

Implications of the MA2 axiom: identification of reasoning domain
The decomposition sub-axiom requires identifying subsystems, parts or even sub-parts which are requested to be externally interfaced: Within these subsystems, parts or sub-parts, actors and their features need to be recognized. In order to recognise system subsystems, it is necessary for an Io analysis to refer to a system model, which can be shared by most of the people belonging to the systems itself.

To provide a sound example, by setting the scope to the value analysis [13], the value chain model of the enterprise decompose it into several parts strategically relevant, satisfying the axioms MA2. Each enterprise could thus be analysed through the activities of the value chain, since these are closer to a physical or logical sequence. Functional view (e.g. business units of departments) would not satisfy an Io analysis better than subsystems identified by areas in the value chain of Porter can do.

Implications of the IA axiom: recognition of interfaces
Let's consider two system models (A and B), made of respectively n (A1, A2, …, An) and m subsystems (B1, B2, …, Bm): if these systems are intended to interoperate, two elements deputed to serve a "symbiosis channel" needs to be identified, where exchanges of information, knowledge, materials or services may take place. It is critical to identify the corresponding elements of the two system deputed to interface. These subsystems could interoperate either with homologous components (e.g., A1-B1) or with heterologous ones (e.g. A1-B2), according to their requirements.

Implications from MA3 sub-axiom: recognition of context
Descending from this sub-axiom, it is evident the fundamental role played by the decision maker's in the perception of reality. Therefore it's necessary to recognise the modelling domain, identifying two sub-parts:

– real part: set of elements perceived by human senses; they are strictly linked to tangible and physical properties;
– virtual part: set of elements perceived by logical senses; they are strictly linked to relational properties.

Implications from SA axiom: recognising critical features
From the systemic axiom, a generic entity can be characterised according to three levels as in Dassisti [4]: a physical level (terminology), a logical level (functional characteristics related to its use, such as properties and attributes) and systemic level (relationship between properties). The existing links between properties and relationships is thus evident: every kind of properties of an entity should be designed considering its relations with external entities.

4 Interoperability Analysis: A Show Case

This section presents an example of a real industrial application made in a company - a big first-tier supplier in the automotive sector. The focus on an internal department, the Study & Research Division (S&RD), and in particular the internal cooperation processes aimed at developing and testing components of a fuel injection system for diesel automotive engines. Given its strategic role, this system had the need to interface with several different departments to close the loop from the design phase to manufacturing one. Even though within the same organisation, divisions and departments had managerial as well as operative autonomy, which made critical the organisation of activities as well as reaching high level performances. For the sake of simplicity here only one external department Production Department (PD) will be considered. In the following table we summarise the features of the show case presented, with the aim to understand its interoperability pertinence.

Table 1. The show case

Iop features	Show case
Entities interact for a temporary common scope	Yes (permanently)
Interaction is needed because of lacking internal resource	Yes (dept organisation)
Entities are heterogeneous	YES
Interaction does not influence autonomy of entities involved	YES
Entities share the same semantic	YES
There is a mediator	YES (hidden)
Cooperation environment is open	NO

A clear heterogeneity exists between the two departments, since their organisation strongly differs. Because share the same norms and procedures of the central units, almost no barriers exists for syntax (but not necessarily of semantic, due to different cultural and the size of the company). The lack of the presence of mediator, to be responsible of exchange, as highlighted in the table, shows a lack of interoperability.

4.1 Scope of Interoperation

According to AA axiom, the first step in the Io analysis was finding pertinent information and documents concerning the product-development process. Most of the information were already collected in a standard document (Product Creation Process) defined by a quality management procedure to assure a high performance of the design process. Design process was structured independently of a specific application as an innovation cycle, and thus it consisted of several sub-steps cantered: Preparation (idea preparation – innovative phase); Project (generic product design – innovative phase); Management; Market Analysis; Acquisition Project (phase with an high interfacing with customers, which ends with orders acquisition); Customer Project (customisation depending on a specific customer requirements). From this organisation, in accordance to the MA1 sub-axiom, it is evident the level of abstraction of activities and thus the different needs for coordination.

The attention is focused on the first project development. This part of the Io analysis aimed to clarify the need to interoperate (WHY), that means the final expected result coming from the temporary joint activity. Essentially, this might descend from a pro-active attitude to reach synergic common objectives (strategic based interoperability) or from a reactive attitude to respond to market/customer solicitation (opportunity based interoperability). According to these two main classes, interoperation scopes were recognised by making explicit the driving forces and resistances within the actors involved.

4.2 Actors Recognition

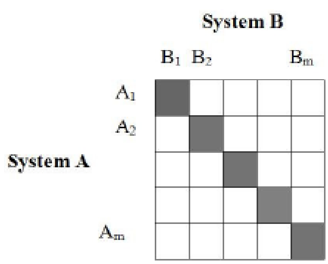
A checklist to evaluate adequate features to interoperate was adopted, to extract adequate information about components involved in common activities (MA1 sub-axiom), as in table 1. Here is reported based on a set of questions put for understanding the attitude only of its craftsmanship.

After a first analysis, the symbiosis channels were identified and recognised, according to IA axiom.

4.3 General Features of Interaction

Referring to IA axiom, at the end of a complete identification of interfaces, an Interoperability matrix of the two generic systems A and B was prepared (as in Table 2). Its elements gave a clear picture of the cooperation's in place between the corresponding sub-parts. The principal diagonal includes information about homologous components (subsystem, parts, sub-parts or actors), the other elements deals with heterologous ones.

Table 2. Example of an interoperability matrix



Such a matrix serves to qualify objects flowing within symbiosis channels and therefore put the basis for a subsequent modelling of systems, as well as for the selection of the most appropriate modelling environment.

The objects exchanged in these channels – through an intranet technology – were: internal procedures, information from a ERP software as well data from a common database managed centrally. Lacking elements recognised were formalization of exchanges, the explicit recognition interfaces and several criticalities of interaction.

4.4 Modelling Interaction

Descending from MA axiom, the Io analysis was then performed with the support of an IEM model [10]. From the model one can find two main exchange flows:

Development Flow and Plant Flow. They were modelled as flows involving resources, since the major part of activities concerns definition, filling and verification of documents. Despite both flows were mostly parallels, they cannot be considered independent, since derived from the same input and contribute to the same output.

According to MA1 sub axiom, a higher level of detail was chosen to analyse all those macro-phases sketched in the two flows; the modelling helped to understand deeply the two processes and essentially the interoperability requirements for an effective working of the whole system.

Performing the modelling forced to find appropriate documentations and information, as well as the interfaces that were not recognised at all. The most of this information was found in standard documents (Product Creation Process) containing prescription about Standard Task (scopes and activities), Participation (roles and actors in the process) and Proofs.

Without going into a deeper detail of the different phases and their models we here briefly summarise the steps related to the scope of Io design:

- Product development: it is pertinence mainly of the Study & Research Division in a strict contact with the customer.
- Product realisation: this phase tends to produce the last prototype before production for final approval and tests.
- Process development: this phase is under the responsibility of the Production department and thus is represented within the Plant Flow of the model developed.
- Process realisation: it is the start-up phase and ends with the delivery of the first products to the customer.
- Ramp-up: this phase is under the responsibility of Study & Research Division where all the design documents and specifications are provided to the customer.

The Io analysis derived the need to improve performance of the strict cooperation between Study & Research Division with Production department, which was almost constant throughout all the phases. Despite belonging to the same department, the complexity of the structures and the number of activities made strong the need to explicitly analyse and improve the interoperation.

The first step of the Io analysis was the definition of resources and their belonging to one or the other department. For instance, Sample design is made either by Production Dept. (Plant) or by Study & Research Division (Design Group, Order Processing).

Several other situations of "apparent overlapping" were recognised thanks to the modelling and Io analysis. These overlapping were identified as interoperations between different components of the two departments, and consequently required a different point of view for the analysis in terms of interoperability.

In the local symbiosis channels mainly products and information were exchanged; referring to the elementary brick - the sequence of activities and test made for building the first prototype - the activity of prototype production is performed by Sample Shop (Production Dept.), while tests on the same prototype are made by Study & Research Division: namely, functional test from the customer care responsible and test endurance tests and force measurements from 'Group of Endurance Test' (sub-division of Study & Research Division). The object B-sample (the first prototype) in thus exchanged thus requiring an interoperable system with adequate efficiency.

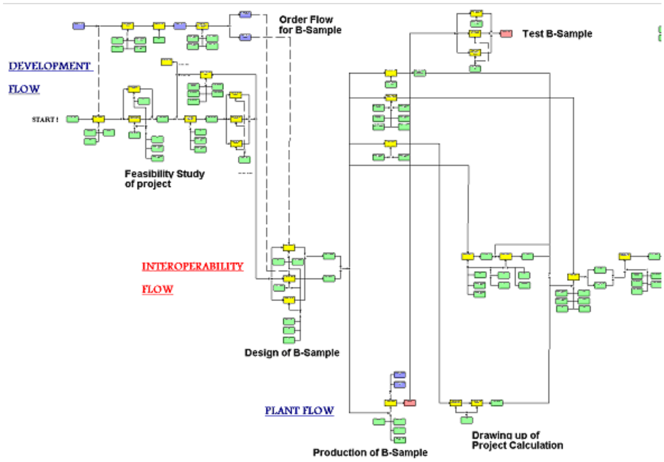


Fig. 2. Section of the whole interoperability model

The Io analysis, according to IA axiom, thus was performed finding those interconnection points in the different flows of products (like the above example) or information, or both, within symbiosis channels building a general interoperability model. Differently from a common AS-IS model, the Io model (see fig.2) tries to represent the product and process development flow, as well as resources devoted to, with the scope to formalise joint activities, objects (either material or immaterial) exchanged or required, the relative flows, so as to understand interfacing requirements (say, synchronisation needs).

Main steps in building this model were: (i) Augmenting the level of detail and granularity; (ii) Recognition of common activities between the two departments; (iii) Identification of Interoperability Flow; (iv) Identification of Development Flow and Plant Flow.

As one can note form figure 2, the Interoperability Flow is located at the centre between the Development Flow (upper) and the Plant Flow (lower): from the analysis of the interactions between the two departments it resulted clear that two main classes of interactions were important to the Io analysis: sharing and exchange of objects (either material or immaterial).

As concerns the interactions requiring sharing of objects, these mainly concerns all those activities that would require, generally speaking, a co-X activities (co-design, co-working, etc.).

Concerning interactions requiring only object exchange, it was clear that mainly required a information exchange of procedural type in this particular case, characteristic of those activities were there is a lower intellectual value-added: say, for instance, quality testing or project verification.

Figure 2 also shows a part of the activities of ‘Process Development’: it is evident the synchronous exchange of information between Product Planner and Study & Research Division: this latter is informed by Production dept. of changes to be performed on the line; on the other hand, Production need a positive feedback from S&R division before proceeding with line preparation.

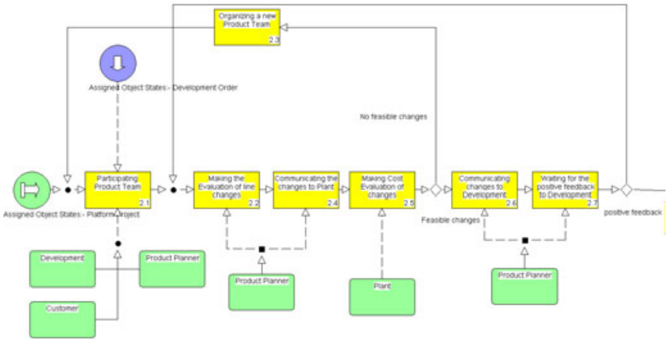


Fig. 3. ‘Example of synchronisation

This kind of interoperation requires synchronisation at different exchange points (interfaces). Thanks to the analysis performed and on the systemic view introduced, within models several interfaces were formalised, thus allowing understanding criticalities of interoperability.

The structure devised from the field for a generic interface was of a set of parallel components (or elements) as here indicated: scope of the interchange; session model; connection management; shared objects. It is here evident how difficult was to model procedures (How of interfacing) that indeed represents the core of interoperation.

Modelling interface, at this stage of analysis, was thus essentially more focused on syntax than on semantic; an interesting point that came out indirectly was the recognition of the critical role of the “mediator” for managing exchanges and regulating operation. In this sense, there was the need to represent more complex functions of the interface than simply an exchange of information. Considering the semantic would thus mean to extend the classical functionalities of software interfaces (reusability, context independence; object orientation, etc.) to a more systemic view of it – SA axiom -: synchronisation or alignment of contents (see fig.3).

5 Concluding Summary

To a general extent, an Io analysis should consist of three main parts: (1) Definition of the scope of interoperation, (2) Recognition of actors and their intrinsic features, (3) Recognition of attitude to interaction (e.g. opportunity, potentiality, and criticality). This paper tentatively presents an interoperability analysis approach which can be used before the design phase of interoperability systems. The approach is guided by a set of axioms, derived on the foundation of the general system theory concepts. The output of this interoperability analysis allows clarifying all those elements useful to derive sound decisions. The approach is elaborated at a high level of abstraction and can thus generically be applied for the analysis of any system.

There are some points to be further investigated so far. First, the interoperability model does not provide definitive solutions on “how” things should be organised to assure interoperability (say, timing, contents, organisation of symbiosis channels, etc.). A second missing point is how to recognise subsystems and actors: this logical decomposition still remains a modeller's subjective process which can influence the subsequent design phase. Finally, it is still not clear if interoperability is an absolute-intrinsic concept (an embedded property) or a relative-contextual to the purpose one.

References

1. Von Bertalanffy, L.: Perspectives on General System Theory. In: Taschdjian, E., Braziller, G. (eds.) New York (1974)
2. Chen, D., Doumeingts, G.: Decisional systems design using GRAI method. In: IA 1992 International Conference on Industrial Automation, Singapore, May 20-23 (1992)
3. Dassisti, M.: Formal modelling approaches for integrated quality-management systems in delocalised enterprises: a taxonomy of requirements. Int. J. Automotive Technology and Management 3(3/4) (2003)
4. Dassisti, M., Scorziello, F.: Enterprise modeling interoperability: a case study of two university laboratories. In: Proceedings of the CIRP ICME 2004, Sorrento, June 30-July 2, pp. 61–65 (2004) ISBN 88-87030-79-0 - Editor Roberto Teti
5. Doumeingts, G., Vallespir, B., Chen, D.: Decision modelling GRAI grid.. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Architectures of Information Systems, pp. 313–337. Springer, Heidelberg (1998)
6. El-Haik, B.: Axiomatic quality: integrating axiomatic design with six-sigma. Wiley-Interscience Publication, U.S.A. (2005) ISBN 0-471-68273-X
7. Forrester, J.W.: Principles of Systems, 2nd edn. Pegasus Communications, Waltham (1968)
8. Goldman, S.L., Nagel, R.N., Preiss, K.: Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer. Van Nostrand Reinhold, New York (1995)
9. IEEE Computer Society, IEEE Standard Glossary of Software Engineering Terminology (1990) E-ISBN: 0-7381-0391-8
10. Mertins, K., Jochem, R.: MO2GO. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Architectures of Information Systems, pp. 589–600. Springer, Heidelberg (1998)
11. Mesarovic, M.D., Macko, D., Takahara, Y.: Theory of hierarchical, multilevel, systems. Academic Press, New York (1970)
12. Mintzberg, H.: The structure of organisations: A synthesis of research. Pretince-Hall (1979)
13. Porter, E.M.: Competitive strategy. Free Press (1982) ISBN 0743260880
14. Simon, E.: The science of the artificial. MIT Press, Cambridge (1969)
15. Suh, N.P.: The Principles of Design. Edition Oxford University Press (1990)
16. Vernadat, F.: Enterprise Modelling and Integration: Principles and Applications. Chapman & Hall, London (1996)
17. Vernadat, F.: Enterprise Modelling and Integration – From fact modelling to Enterprise Interoperability. In: Kosanke, K., et al. (eds.) Enterprise Inter-and-Intra Organisational Integration, Building International Consensus. Kluwer Academic Publishers (2003)

Semantic Annotation Model Definition for Systems Interoperability

Yongxin Liao[1], Mario Lezoche[1,2], Hervé Panetto[1], and Nacer Boudjlida[2]

[1] Centre de Recherche en Automatique de Nancy (CRAN), Nancy-University, CNRS, BP 70239,54506 Vandoeuvre-lès-Nancy Cedex, France
{Yongxin.Liao,Mario.Lezoche,Herve.Panetto}@cran.uhp-nancy.fr
[2] LORIA, Nancy-University, CNRS,BP 70239, 54506 Vandoeuvre-lès-Nancy Cedex, France
Nacer.Boudjlida@loria.fr

Abstract. Semantic annotation is one of the useful solutions to enrich target's (systems, models, meta-models, etc.) information. There are some papers which use semantic enrichment for different purposes (integration, composition, sharing and reuse, etc.) in several domains, but none of them provides a complete process of how to use semantic annotations. This paper identifies three main components of semantic annotation, proposes for it a formal definition and presents a survey of current semantic annotation methods. At the end, we present a simple case study to explain how our semantic annotation proposition can be applied.The survey presented in this paper will be the basis of our future research on models, semantics and architecture for enterprises systems interoperability during the product lifecycle.

Keywords: Semantic Annotation, Models, Ontology, Systems Interoperability.

1 Introduction

Nowadays, the need of systems collaboration across enterprises and through different domains has become more and more ubiquitous. But because the lack of standardized models or schemas, as well as semantic differences and inconsistencies problems, a series of research for data/model exchange, transformation, discovery and reuse are carried out in recent years. One of the main challenges in these researches is to overcome the gap among different data/model structures. Semantic annotation is not only just used for enriching the data/model's information, but also it can be one of the useful solutions for helping semi-automatic or even automatic systems interoperability.

Semantically annotating data/models can help to bridge the different knowledge representations. It can be used to discover matching between models elements, which helps information systems integration [1]. It can semantically enhance XML-Schemas' information, which supports XML documents transformation [2]. It can describe web services in a semantic network, which is used for further discovery and composition [3]. It can support system modellers in reusing process models, detecting cross-process relations, facilitating change management and knowledge transfer [4]. Semantic annotation can be widely used in many fields. It can link specific resources according to its domain ontologies.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 61–70, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The main contribution of this paper is identifying three main components of semantic annotation, giving a formal definition of semantic annotation and presenting a survey, based on the literature, of current semantic annotation methods that are applied for different purposes and domains. These annotation methods vary in their ontology (languages, tools and design), models and corresponding applications.

The remaining of this paper is organized as follows: Section 2 describes the definition of annotation and gives a formal definition of semantic annotations. Section 3 first provides the answers to why and where to use semantic annotation, then introduces the languages and tools that can be used to develop ontology, the design of semantic annotation structure models and the corresponding applications. Section 4 describes a simple case study example. Section 5 concludes this paper, together with some related work and potential extensions.

2 What Is Semantic Annotation?

In Oxford Dictionary Online, the word "annotation" is defined as "*a note by way of explanation or comment added to a text or diagram*". It is used to enrich target object's information, which can be in the forms of text descriptions, underlines, highlights, images, links, etc. Annotation has special meanings and usages in different fields. In java programming, annotation can be added on classes, methods, variables, parameters, etc., for example, JUnit[1] is a test framework that is based on java annotation. In mechanical drawing, an annotation is a snippet of text or symbols with specific meanings. In Library Management, an annotation is written in a set form (numbers, letters, etc.), which helps the classification of books.

Further, different annotation types are identified by [5] and [6]. They distinguished annotation as (i) *Textual annotation*: adding notes and comments to objects; (ii) *Link annotation*: linking objects to a readable content; (iii) *Semantic annotation*: that consists of semantic information (machine-readable). Similarly, three types of annotation are described in [7]: (i) *Informal annotation*: notes that are not machine-readable; (ii) *Formal annotation*: formally defined notes that are machine-readable (but it does not use ontology terms); (iii) *Ontological annotation*: notes that use only formally defined ontological terms (commonly accepted and understood).

According to the above classification, semantic annotation can be considered as a kind of formal metadata, which is machine and human readable.

2.1 Semantic Annotation

The term "Semantic Annotation" is described as "*An annotation assigns to an entity, which is in the text, a link to its semantic description. A semantic annotation is referent to an ontology*" in [3]. Semantic annotation is concerned as "*an approach to link ontologies to the original information sources*" in [8]. These definitions from different papers show one thing in common: a semantic annotation is the process of linking electronic resource to a specific ontology. Electronic resource can be text contents, images, video, services, etc. Ontology here is only one of the possible means to provide a formal semantic.

[1] http://www.junit.org/

As it can be seen on Figure 1, the left side represents an Electronic Resource (ER) and on the right side, there are the three main components of semantic annotation: (1) *Ontology*, which defines the terms used to describe and represent a body of knowledge [21]. (2) *Semantic Annotation Structure Model (SASM)*, which organizes the structure/schema of an annotation and describes the mappings between electronic resources and one, or more, ontologies. (3) *Application*, which is designed to achieve the user's purposes (composition, sharing and reuse, integration, etc.) by using SASM. This figure also shows the three main steps on how to use semantic annotation, which is introduced in section 3.

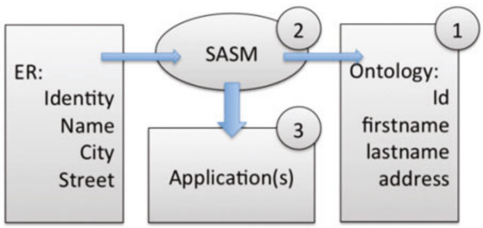


Fig. 1. Semantic Annotation components

2.2 Formal Definition of Semantic Annotation

The following definition formally defines a semantic annotation: a *S*emantic Annotation SA is a tuple $(\mathcal{M}, \mathcal{A})$ consisting of the SASM $\mathcal{M}$ and an application $\mathcal{A}$.

$$SA := \{\mathcal{M}(\mathcal{E}, \mathcal{P}(\mathcal{O})), \mathcal{A}\}$$

Where:

$\mathcal{O} = \{o_1, o_2, \dots, o_n\}$, is the set of ontology o_i that bring some meaning to any annotated element.

An *Ontology* $o_j \in \mathcal{O}$ is a 4-tuple (C_{o_j}, is_a, R_{o_j}, σ_{o_j}), where C_{o_j} is a set of *concepts*, *is_a* is a partial order relation on C_{o_j}, R_{o_j} is a set of relation names, and $\sigma_{o_j}: R_{o_j} \rightarrow (C^+)$ is a function which defines each relation name with its arity [9].

Formally, $\mathcal{M} = \{m_x : \langle e_i, p_j \rangle \mid e_i \in \mathcal{E} \times p_j \in \mathcal{P}(\mathcal{O})\}$represents the set of relationships between an element e_i of the set of electronic resources $\mathcal{E}$ and an element p_j of the powerset of the ontology set $\mathcal{O}$.

$m_x(e_i, p_j)$ is a binary relationship that maps e_i to p_j in the context of an application $\mathcal{A}$. It may represent four different kinds of semantic relations:

(1) $m_{\sim}(e_i, p_j)$: stating that e_i is semantically *equivalent* to p_j.
(2) $m_{\supset}(e_i, p_j)$: stating that e_i *subsumes* the semantic of p_j.
(3) $m_{\subset}(e_i, p_j)$: stating that e_i *is subsumed by* the semantic of p_j.
(4) $m_{\cap}(e_i, p_j)$: stating that e_i *intersects* with the semantic of p_j.

$\mathcal{M}$ can be further extended, including also some additional parameters or constraints c_k, generally expressed using, in the worst case, natural language, or, better, a formal logical expression. $\mathcal{M}$ is then defined as $\mathcal{M} \coloneqq \{m_x, c_k\}$.

The main issue, related to mappings such as in (2) and in (3), is being able to measure the semantic gap (2) or the over semantic (3), brought by the semantic annotation. Such measures have been studied by researchers in the domain of information retrieval [10] or in the domain of ontology merging [9], matching[11], mapping [12] and alignment[13].

In addition, [14] also gave a very simple definition of semantic annotation which is $SA := (R, O)$, where R is set of resources and O is an ontology. Furthermore, [15] defined it as $SA := \{R_A, C_A, P_A, L, T_A\}$. In this definition, R_A is a set of resources; C_A is a set of concept names; P_A is a set of property names; L is a set of literal values; and T_A is a set of triple (s, p, v), where $s \in R_A, p \in P_A, v \in (R_A \cup L)$. To the best of our knowledge, T_A in this definition is duplicated.

3 Why, Where and How to Use Semantic Annotation?

Semantic annotation uses ontology objects to enrich resource's information that tells a computer the meanings and relations of the data terms. It can be used to bridge the gap between modelsas additional information that helps description, discovery and composition.

Semantic Annotations are generally used in heterogeneous domains.We found several papers presenting different employments. [1] used semantic annotations to annotate the model/object at CIM, PIM and PSM levels of the MDA approach [17], which helps information system integration. In the research of [2], a path expression method is developed for adding annotations to XML-Schemas. Then they transform paths to ontology concepts and use them to create XML-Schema mappings that help XML document transformation. [3] used it to help matching and composition algorithm, which represents web services as a semantic network and produces the best composition plan. [4] used semantic description of process artefacts to help graphical modelling of business processes. In the research of [8], a semantic annotation framework is designed to manage the semantic heterogeneity of process model, to solve the discovery and sharing of process models problems in/between enterprise(s). In [16], an annotation framework is proposed for semi-automatically marking up web service descriptions (WSDL files) with domain ontologies to help web services discovery and composition. In the next three subsections we will present our proposed three semantic annotation components.

3.1 Step 1: Design or Select Ontology

Design or select an appropriate ontology for semantic annotations is the first step of the annotation process. Ontology has been actively studied for a long period of time, and there are many research works proposing ontology-engineering techniques, such as Ontolingua[2], F-logic[3], OWL[4], etc. We are not going to give, here, a complete

[2] http://www.ksl.stanford.edu/software/ontolingua/

[3] http://flora.sourceforge.net/

[4] http://www.w3.org/TR/owl-features/

overview of every ontology languages, but we provide a brief introduction toseveral examples.

[1] designed a SMM (Semantic Meta-model) to describe domain ontologies. Artefacts in ontology are castigated as DomainFunction and DomainObject. The relations (predicates) among Objects and Functions are defined as: *IsA*, *IsInputOf*, *IsOutputOf*, *Has*, etc. A RDF-like triple (e.g., *Tax Has TaxNumber*) is used as the statement in SMM.

[4] used two kinds of ontologies: sBPMN[5] ontology and a domain ontology. The first ontology is used to represent BPMN process models. The second ontology defines domain objects, states and actions according to objects lifecycle, which is used to provide the user advices during the modelling process.

[8] used Protégé OWL editor to design the ontology. In order to separately annotate meta-models (modelling language) and their process models, the author designs two ontologies: *General Process Ontology (GPO)* and *Domain Ontology*. The design of GPO is based on Bunge-Wand-Weber (BWW) Ontology [18]. The Domain ontology is formalized according to SCOR[6] specifications (Supply Chain Operations Reference-model).

3.2 Step 2: Design the Semantic Annotation Structure Model

The second component of a semantic annotation is SASM. It is the connection between electronic resources, applications and ontology concepts. A study in this direction is pursued by SAWSDL Working Group[7] that developed SAWSDL [19] (Semantic Annotation for Web Services Definition Language) which provides two kinds of extension attributes as follow: (i) *modelReference*, to describe the association between a WSDL or XML Schema component and a semantic model concept; (ii) *liftingSchemaMapping* and *loweringSchemaMapping*, to specify the mappings between semantic data and XML [20].

To be more specific, we abstract four structure models that are designed for different requirements. Figure 2 below gives an overview of these three SASMs: Model A is proposed to conceptually represent a web service from [3]; Model B is designed to annotate the business process model from [4]; and Model C is the annotation model for an activity element which is part of the Process Semantic Annotation Model (PSAM) from [8].

In order to compare above semantic annotation structure models, we identify five types for classifying the contents in SASM:

(1) *identity* of annotation;
(2) *reference to ontology concept*;
(3) *reference to element* (represent the relationship between element themselves);
(4) *text description*, the natural language definitions of annotation contents;
(5) *others* (extinction contents, such as: execution time, restriction, etc.).

[5] http://www.ip-super.org
[6] http://supply-chain.org/
[7] http://www.w3.org/2002/ws/sawsdl/#Introduction

We can easily find that the basic components of SASMs are: *identity of annotation* and *reference to ontology concepts*; *reference to element*, *text description* and *others* are added for different usages. As examples, [3] added “exec-time” into SASM to record the execution time of a web service request and used “inputs” and “outputs” to represent the relationships between processes; [4] described the references with meaning of states of objects (current, before and after); [8] adds “has_Actor–role” to denote the relationship between activity element and actor-role element.

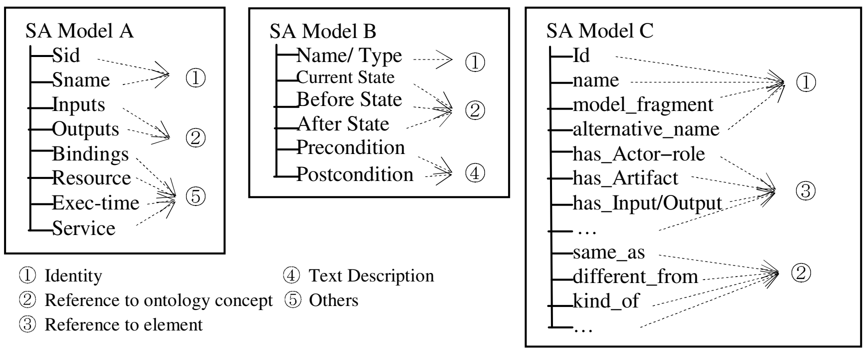


Fig. 2. Semantic Annotation Structure Model Examples

Furthermore, one to one mapping is not the only mapping type in SASM. For example, in Model A, there can be more than one input, which means the mapping between model content and ontology concept is one to many. Here, we analyses the mappings of “*reference to ontology concepts*”.

Mappings are separated into two levels in the research of [8]: meta-model level and model level. In the meta-model level the mappings are defined as: *Atomic Construct*, *Enumerated Construct* and *Composed Construct*. In model level, semantic relationships are: *Synonym*, *Polysemy*, *Instance, Hyponym*, *Meronym*, *Holonymand Hypernym*. Five mapping types are described in [1]: *single representation*, *containment*, *compositions*, *multiple* and *alternative representation*.

In our opinions, there are three high level mapping types: *1 to 1* mapping, *1 to n* mapping and *n to 1* mapping (n to n is a combination of 1 to n and n to 1). For each of the mapping, we can design different semantic relationships for further usages. In *1 to 1* mapping semantic relationships can be like: “equal_to”, “similar_to”, etc. In *1 to n* they can be as: “contains”, “has”, etc. In *n to 1* they can be as: “part_of”, “member_of”, etc. One element can have several semantic relationships, but for each relationship, they belong to one mapping type.

Since the structure and semantic relationships are designed, we should consider how to implement the annotation process. It can be performed manually, semi-automatically or automatically. In the research of [8], mapping is manually linking the process models to ontology. [16] developed algorithms to semi-automatically match and annotate WSDL files with relevant ontologies. Automatic mapping is, for the moment, restricted to some simple cases because of the impossibility to completely explicit knowledge from the different models.

3.3 Step 3: Develop the Application

Once SASM step is accomplished, designers can begin to design the application to achieve their purpose (composition, sharing and reuse, integration, etc.). In several different domains semantic annotation methods are used to resolve particular problems.

In the domain of web services, [3] presented a matching algorithm to process the “input” and “output” (SASM model C, Figure 3) of elements, and builds a semantic network. This network is explored by a composition algorithm, which automatically finds a composite service to satisfy the request;

In business process modelling domain, [4] designed name-base and process context-base matchmaking functionalities to help user annotating process models. Name-base matching uses string distance metrics method for the matching between business process models and domain ontology. Process context-base matching uses the lifecycle (state before, state after, etc.) in domain ontology for suggesting the next activity during modelling.

[8] developed a prototype Process Semantic Annotation tool (Pro-SEAT), which is used to describe the relationship between process models and ontologies. They use Metis[8] as a modelling environment integrating Protégé OWL API to provide an ontology browser. Ontologies are stored on an ontology server, which can be loaded by annotators. The output of the annotation is an OWL instance file that is used to support the process knowledge query, discovery and navigation from users.

Indeed, there are many tools and technologies that enable designing applications in semantic annotation. The selections are always depending on the design of SASM and ontology. In any case, all three components of semantic annotation are closely related.

4 A Simple Case Study

To explain how our semantic annotation proposition can be applied, we give a simple example to illustrate the annotation process between Sage X3[9], an Enterprise Resource Planning (ERP) model and Flexnet[10], a Manufacturing Execution System (MES) model focused on Bill of Materials concept.

As can be seen in Figure 3, the ONTO-PDM product ontology, based on Tursi work [22], is used by both systems as the annotation source. In general cases two systems use different ontologies and the solution can be: mapping between each other; mapping with a reference ontology. Our solution provides SASM and its two main contents are “①identity of annotation”, “②reference to ontology concept”. The annotate process is performedmanually by a domain expert. Element “BillOfMaterialsID” in Sage X3 model is annotated as semantic annotation SA_1, which is referenced to the ontology concept “MaterialEntity”; Element “BomNumber” in Flexnet model is annotated as semantic annotation SA_2, which is referenced to the same ontology concept “MaterialEntity”.

[8] http://www.troux.com/
[9] http://www.sage.com
[10] http://www.apriso.com/products/flexnet_production/

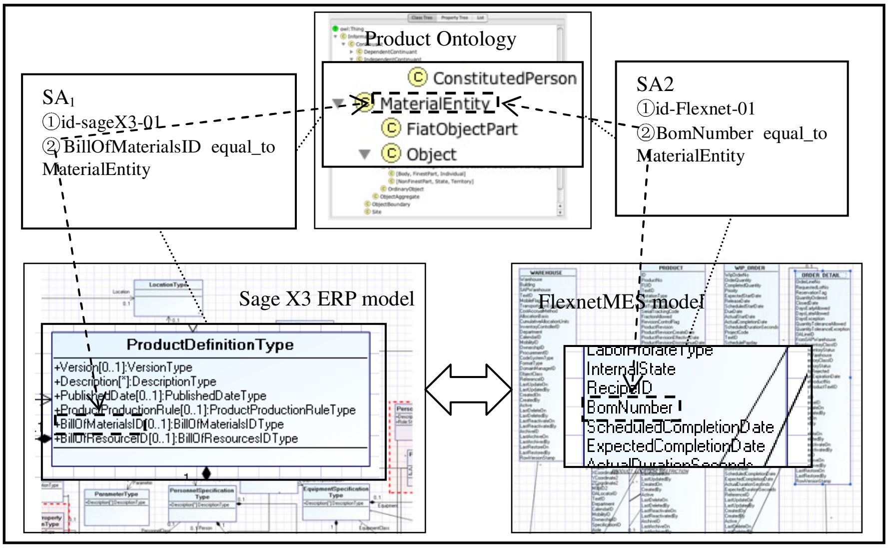


Fig. 3. Semantic Annotation between Sage X3 and Flexnet

Semantic Annotation SA_1 can be formally described as $SA_1 = \{m_{\sim}(ProductDefinitionType.BillOfMaterialsID, MaterialEntity), id_1, A\}$ (id_1 is the identity parameter that refers to"id-sageX3-01"). Similarly, SA_2 can be formally described as $SA_2 = \{m_{\sim}(WIP\_Order.BomNumber, MaterialEntity), id_2, A\}$ .Through the comparisonof SA_1 and SA_2, application A can match the two different models concepts. However, the design of our semantic annotation model definition still needs to be further developed. We will discuss the possible problems and future work in the last section.

5 Conclusions

In this paper, a brief survey of semantic annotation in different domains is presented. We identify three main components of semantic annotations that are Ontology, SASM and Application. In addition, a formal definition of semantic annotation is proposed. It contributes to better understand what a semantic annotation is and proposes a common reference model. But, how using semantic annotation? There are still many problems can be further discussed during the annotation process. For example, how to optimize ontology and an annotated model? How to solve the inconsistency or conflicts during the mapping? How to add consistent semantic on models in different levels of a system? How to achieve semi-automatic or automatic annotation?

We are currently investigating how semantic annotations can help collaborative actors (organizations, design teams, system developers, etc.) in co-designing, sharing, exchanging, aligning and transforming models. In particular, this research work will be based on general systems with several kinds of interactions. We can have interoperation between systems that with different versions (during many years, systems may have been modified or updated). We can also have systems with same functions but used by different enterprises. Semantic annotations can bridge this knowledge gap and identify differences in models, in schemas, etc. In some case, interoperation is a process between a set of related systems throughout a product lifecycle (Marketing, Design, Manufacture, Service, etc.), and semantic annotations can influence the existing foundations and techniques which supports models reuse, semantic alignment and transformation, etc. Above all, our research work will focus on designing, and reusing appropriate ontologies in relationship with a formal semantic annotation structure model.

References

1. Agt, H., Bauhoff, G., Kutsche, R., Milanovic, N., Widiker, J.: Semantic Annotation and Conflict Analysis for Information System Integration. In: Proceedings of the 3rd Workshop on Model-Driven Tool & Process Integration, pp. 7–18 (2010)
2. Köpke, J., Eder, J.: Semantic Annotation of XML-Schema for Document Transformations. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 219–228. Springer, Heidelberg (2010)
3. Talantikite, H.N., Aïssani, D., Boudjlida, N.: Semantic annotations for web services discovery and composition. Computer Standards & Interfaces 31(6), 1108–1117 (2009)
4. Born, M., Dörr, F., Weber, I.: User-Friendly Semantic Annotation in Business Process Modeling. In: Weske, M., Hacid, M.-S., Godart, C. (eds.) WISE Workshops 2007. LNCS, vol. 4832, pp. 260–271. Springer, Heidelberg (2007)
5. Bechhofer, S., Carr, L., Goble, C., Kampa, S., Miles-Board, T.: The Semantics of Semantic Annotation. In: Proceedings of the 1st International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems, pp. 1151–1167 (2002)
6. Boudjlida, N., Dong, C., Baïna, S., Panetto, H., Krogstie, J., Hahn, A., Hausmann, K., Tomás, J.V., Poler, R., Abián, M.Á., Núñez, M.J., Zouggar, N., Diamantini, C., Tinella, S.: A practical experiment on semantic enrichment of enterprise models in a homogeneous environment. INTEROP NoE Deliverable DTG4.1. INTEROP NoE IST 508011 (2006), doi:http://www.interop-vlab.eu
7. Oren, E., HinnerkMöller, K., Scerri, S., Handschuh, S., Sintek, M.: What are Semantic Annotations? Technical report, DERI Galway (2006)
8. Lin, Y.: Semantic Annotation for Process Models: Facilitating Process Knowledge Management via Semantic Interoperability. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway (2008)
9. Stumme, G., Maedche, A.: Ontology Merging for Federated Ontologies on the Semantic Web. In: Proceedings of the International Workshop for Foundations of Models for Information Integration (2001)
10. Ellis, D.: The Dilemma of Measurement in Information Retrieval Research. Journal of the American Society for Information 47(1), 23–36 (1996)
11. Maedche, A., Staab, S.: Measuring Similarity between Ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 251–263. Springer, Heidelberg (2002)
12. Doan, A., Madhavan, J., Domingos, P., Halevy, A.Y.: Learning to map between ontologies on the semantic web. In: Proceedings of the World Wide Web Conference, pp. 662–673 (2002)

13. Noy, N.F., Musen, M.A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proceedings of the 17th National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 450–455 (2000)
14. Peng, W., Baowen, X., Jianjiang, L., Dazhou, K., Yanhui, L.: A Novel Approach to Semantic Annotation Based on Multi-ontologies. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics, vol. 3, pp. 1452–1457 (2004)
15. Luong, P., Dieng-Kuntz, R.: A Rule-based Approach for Semantic Annotation Evolution. Computational Intelligence 23(3), 320–338 (2007)
16. Patil, A., Oundhakar, S., Sheth, A., Verma, K.: Meteor-S Web Service annotation framework. In: Proceedings of the 13th International Conference on the World Wide Web, pp. 553–562 (2004)
17. Mellor, S.J., Scott, K., Uhl, A., Weise, D.: Model-Driven Architecture. In: Bruel, J.-M., Bellahsène, Z. (eds.) OOIS 2002. LNCS, vol. 2426, pp. 290–297. Springer, Heidelberg (2002)
18. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Information System Journal 3(4), 217–237 (1993)
19. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: SAWSDL: Semantic Annotations for WSDL and XML Schema. IEEE Internet Computing 11(6), 60–67 (2007)
20. Martin, D., Paolucci, M., Wagner, M.: Towards Semantic Annotations of Web Services: OWL-S from the SAWSDL Perspective. In: Proceedings of the OWL-S Experiences and Future Developments Workshop at ESWC 2007 (2007)
21. Boyce, S., Pahl, C.: Developing Domain Ontologies for Course Content. Educational Technology & Society, 275–288 (2007)
22. Tursi, A., Panetto, H., Morel, G., Dassisti, M.: Ontological approach for Products-Centric Information System Interoperability in Networked Manufacturing Enterprises. IFAC Annual Reviews in Control 33(2), 238–245 (2009)

Monitoring Morphisms to Support Sustainable Interoperability of Enterprise Systems

José Ferreira[1], Carlos Agostinho[1,2], João Sarraipa[1,2], and Ricardo Jardim-Goncalves[1,2]

[1] Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
jaf14954@fct.unl.pt
[2] Centre of Technology and Systems, CTS, UNINOVA, 2829-516 Caparica, Portugal
{ca,jfss,rg}@uninova.pt

Abstract. Nowadays, organizations are required to be part of a global collaborative world. Sometimes this is the only way they can access new and wider markets, reaching new opportunities, skills and sharing assets, e.g. tools, lessons learnt. However, due to the different sources of enterprise models and semantics, organizations are experiencing difficulties in exchanging information electronically and seamlessly. To solve this issue, most of them try to attain interoperability by establishing peer-to-peer mapping morphisms with different business partners, or in optimized networks use neutral data standards to regulate communications. Moreover, more and more enterprise systems are dynamic, looking forward to meet further requirements, causing new interoperability problems and efficiency reduction. This paper proposes a multi-agent framework to monitor existing enterprise system networks, being capable of detecting morphism changes. An example from systems engineering is presented, where network harmonization breakings are timely detected, and possible solutions are suggested to regain the interoperable status, thus enhancing robustness for reaching sustainability of business networks.

Keywords: Interoperability, Model Morphisms, Sustainable Interoperability, Multi Agent Systems, Model-Based Systems Engineering.

1 Introduction

In the last few years, collaboration between different enterprises has increased significantly with the possibility of combining resources towards achieving a common goal, as well as to boost productivity and reduce expenses. This increases the chances of survival of smaller enterprises in the current turbulent market [1].

Schrage [2], emphasizes that the issue in collaboration *"is the creation of value"*. In this context, not only data exchange in industrial processes (e.g. supply chain, e-procurement, etc.), but also in systems engineering (SE) has been a target of multiple collaborative networks. It demands the combination of multiple complex processes, such as requirements elicitation, product design and development, teams' management, logistics, etc, that normally are performed in conjunction by different enterprises. The systems engineering method recognizes that each system is an

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 71–82, 2011.
© Springer-Verlag Berlin Heidelberg 2011

integrated whole composed of diverse, specialized structures and sub-functions that aid in the development and life cycle (LC) of a product, using information and product models to describe and integrate the whole process [3].

In this sense, being defined as the ability that two or more systems have to exchange information and use it accurately [4], interoperability, namely the lack of it, could disturb the functionalities of the enterprise systems networks and decrease their competitiveness and innovation. When applications are used to manage the same or different models within a network, several types of costs could be incurred regarding interoperability, like translation or data re-entry for seamless information flows [5]. This situation has a direct impact in the business processes and SE practices that cross-cut these organizations (e.g. collaborative product development), since it is essential to have interoperable models and efficient data exchange for the integrated design, product breakdown (decomposition into parts) and manufacturing [6].

However, in our days, when achieved, this stability is hard to maintain since due to market requirements and the increasingly dynamicity of customer needs, business requirements are constantly being adapted, causing systems, models and semantics to change, thus leading to harmonization breaking and disruption of the interoperability in the enterprise systems networks [7]. As suggested by Agostinho et al. [5], a possible solution for this problem relies on the modelling of data, semantic and structural mappings which enable interoperable relationships and cooperation, as traceable tuples, that, when integrated into knowledge bases dedicated to managing mismatches during communications, can be closely monitored and reactions to network evolutions triggered automatically.

Based on this approach, this paper contributes proposing a new framework to support sustainable interoperability of systems, as introduced by Agostinho and Jardim-Goncalves on [7]. After present in sections 2 and 3 the state of the art in methods and technology for model-based engineering and model-morphisms, the paper proposes in section 4 a multi-agent system to monitor existing enterprise systems, being capable of detecting requirements for morphism evolutions and proposing proper adaptations to restore and reach interoperability on a collaborative network. Section 5 illustrates a SE case study to validate results. Finally in Section 6, the authors present conclusions.

2 Model-Based Systems Engineering

Model-based systems engineering (MBSE) is the formalized application of modelling to support the systems engineering processes, namely requirements, design, analysis, verification and validation activities, beginning at the conceptual design phase and continuing throughout development and later LC stages [8], [9].

Among the many ways to describe the LC of a product or a system, some are more focused on design and production, whilst others are centred on support. Recently, the disposal stages have also been carefully studied due to the impact in the environment [5], [10]. Fig. 1 illustrates a possible view, merging the development process, starting from the concept where the stakeholder's needs are identified until the disposal of the product, with some types of information models used in MBSE, namely [11]:

- The *requirements models* that represent the relationships between user requirements and/or model objects. A primary benefit of modelling requirements is the opportunity this provides for analyzing them with techniques such as requirements animation, reasoning, etc [12];
- The *behaviour models* to represent the intended and unintended behaviours for a system of interest (e.g. a product), thus responding to functional requirements;
- The *parametrics models* to reply to the non-functional requirements representing the formal relationships and constraints of the system and its components.
- And finally the *structure models* which describe the enterprise and system level contexts from both the logical and physical viewpoints.

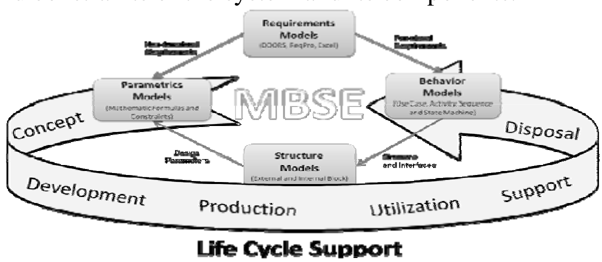


Fig. 1. MBSE scope (adapted from [11], [14])

With MBSE, there is the opportunity to address many of the limitations of the traditional document-based approach, by providing a more rigorous means for capturing and integrating requirements, design, analysis, and verification information, as well as facilitating the maintenance, assessment, and communication of this information across the system's LC. In a nutshell, MBSE can provide *Enhanced communications*, offering more understanding between the development teams and the other stakeholders; *Reduced development risk*, because requirements validation and design verification improve the cost-effectiveness in the development of a system; *Improved quality*, with a more accurate traceability between requirements, design, analysis, and testing; *Increased productivity*, having an more flexible and easier readjustment of the existing models; and *enhanced knowledge transfer*, since all partners involved in the LC stages share the same models [14].

By the exposed, if properly integrated, MBSE framework can bring an added value to SE enterprise networks, maximizing the efficiency of collaborations and stimulating interoperability via the models used along the system's and products LC. However, the issues on robustness and sustainability raised by the dynamicity of networks and evolution of requirements remain pertinent and unaddressed so far.

2.1 Requirements on Modelling

Among the multiple methods to model a system or a product, there is no absolute recipe to develop it. Nevertheless, modelling using MBSE requires having the requirements properly formalized so that the remaining models can be specified with a good level of detail and the final product can achieve the expected quality. Thus, Ogren [15] defines 5 key principles on modelling, namely: *Determinism with formality*, so that everything expressed in the model has a single, defined and obvious meaning; *Understandability*, since the models are only useful is they are readily understood, without extensive education or experience in software or mathematics; *Inclusion of system missions* to be able to extract the systems missions out the models and express how different parts of the system contribute together; *Modelling of structure and behaviour*, to support splitting a system into subsystems; And the *possibility of verification support,* to validate a model against requirements, completeness, etc.

2.2 Information Models and Data Standards for Interoperability

For the description of the information models of Fig.1, MBSE framework suggests the use one of two standard languages, i.e. SysML and AP233, to help in the creation of system engineering models. The first is a graphical modelling language from the Object Management Group (OMG) that supports the analysis, specification, design, verification and validation of complex systems [15]. The language was created to specify and architect systems and its components. With it, it is possible to integrate with UML for software design and VHDL for hardware design. The second is a standard Application Protocol (AP) under STEP initiative [16]. ISO 10303 (STEP) is an international standard for the computer-interpretable representation of product information and for the exchange of product data. Its objective is to provide a means of describing product data throughout its LC [10]. STEP APs are described using EXPRESS [17], the STEP modelling language, which combines methods from the entity-attribute-relationship languages with object modelling concepts. EXPRESS provides general and powerful mechanisms for representation of inheritance among the entities constituting the AP standard model, and it also encloses a full procedural programming language used to specify constraints on populations of instances.

However, despite having these two standards associated to MBSE, frequently enterprises remain using traditional models for engineering analysis and simulation models such as CAD, CAE, reliability, costing, PLM. This raises an interoperability problem to MSBE collaborative networks, since without using SysML or AP233 to integrate the four models needed in SE, the study of how models interact with each other throughout the system LC to benefit stakeholders remains an open issue. In fact, the MSBE Modelling and Simulation Interoperability (MSI) Challenge Team is active looking for a solution to incorporate the diversity of models currently in use in SE [18]. So far, the most common way to integrate all models continues to rely on P2P mapping morphims, similarly to what exists on inter-enterprise communications [7].

3 Models Morphisms (MoMo)

A morphism is described in mathematics as an abstraction of a structure-preserving mapping between two mathematical structures [19]. This concept is gaining momentum in computer science, more exactly in systems interoperability where it describes the relations (e.g. mapping, merging, versioning, transformation) between two or more system model specifications, denominated model-morphism [20].

As an example of a model transformation based on a morphism, Fig. 2 illustrates a morphism where the “firstName” and “lastName” of a “Person” in an information model are merged to create the “personName” of a different model for “Person”.

Nonetheless, the actual research work conducted in MoMo has been focused on the mappings among ontological structures, as evidenced in Fig. 3. Indeed, Ehrig and Staab [21] describe mapping as the relationship each entity (concept C, relation R, or instance I) in ontology O1 has with a corresponding entity with the same intended meaning, in ontology O2. In the example of Fig. 3, both ontologies are used to describe a person, but they have different mismatches associated with the entities that represent a person: one is about the name that is divided into first and last name in the second ontology; the other is about the weight which is not very precise in the second ontology (i.e. qualitative instead or quantitative weight).

Fig. 2. Example of Transformation

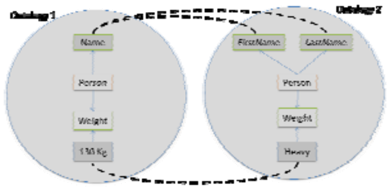
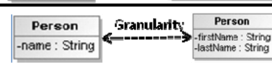
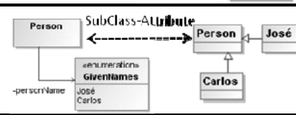
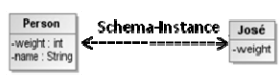
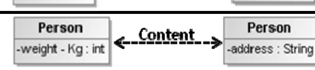
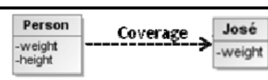
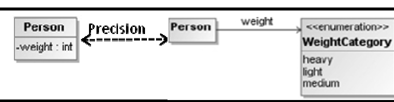


Fig. 3. Example of Mapping between two Ontologies

Table 1. Semantic Mismatches (based on [19])

Mismatch		Description	Examples
Lossless	Naming	Different labels for same concept or structure	
	Granularity	Same information decomposed (sub)attributes	
	Structuring	Different design structures for same information	
	SubClass-Attribute	An attribute, with a predefined value set represented by a subclass hierarchy	
	Schema-Instance	An attribute value in one model can be a part of the other's model schema	
	Encoding	Different formats of data or units of measure	
Lossy	Content	Different content denoted by the same concept	
	Coverage	Absence of information	
	Precision	Accuracy of information	
	Abstraction	Level of specialisation	

3.1 Semantic Mismatches

When a mapping is created between two models, sometimes some inconsistencies of information will appear derived from the multiple conflicts between entities. Those are called semantic mismatches and can be classified as either lossy or lossless, as shown in Table 1. With lossless mismatches, the relating element can fully capture the semantics of the related while in the lossy mismatches a semantic preserving mapping to the reference model cannot be built [19].

3.2 Knowledge Enriched Tuple for Mappings Representation

Due to the mismatches, morphisms should be represented using formal expressions and stored on dedicated knowledge bases, facilitating a sustainable interoperability through the creation of intelligent systems able to reason, deduce and recommend mapping readjustments, as well as transform the mismatch type when system requirements change [7], [22].

MoMo introduced a method of describing relationships/transformations among models [23]. Originally graph theory has been used, although other theories can be considered to achieve the envisaged goals, e.g., set theory [24], model management [25], or semantic matching as explained before. However, none present a perfect solution that can be used to achieve all the goals at once. Some are ideal for structural issues, others for semantics providing good human traceability, while others have a more formal basis. Thus, for formalization purposes, the authors propose in [4] a combination of different strategies in a 5-tuple mapping expression (equation 1):

Mapping Tuple (MapT): <ID,MElems,KMType,MatchClass,Exp> (1)

- *ID* is the unique identifier of the MapT;
- *MElems* is the pair (a,b) that indicates the mapped elements;
- *KMType* stands for Knowledge Mapping Type, and can be classified as: “Conceptual” if mapping concepts and terms; “Semantics” if mapping model schemas; and “InstantiableData” if the mapping is specifying instantiation rules.
- *MatchClass* stands for Match/Mismatch Classification and depends on KMType. Some of the mismatches used are the ones presented in table 1;
- *Exp* stands for the mapping expression that translates and further specifies the previous tuple components.

Concerning the knowledge storage aim, Sarraipa et al. [26] proposed a knowledge base (KB) for this tuple’s information, and suggested that all the business partners in a collaborative network should have a KB (hereafter designated as Communication Mediator - CM) in their local system. CM is built as an extension to the Model Traceability Ontology defined in [27] to act as a mediator for information exchange.

4 Monitoring Morphims for Sustaining Interoperability in MBSE

In a system, interoperation should be addressed at different levels. Thus, the diversity, heterogeneity, and autonomy of software components, application solutions, business processes, and the business context of an enterprise must be considered. Following that trend, interoperability's definition has been revisited a number of times. In Enterprise Interoperability (EI), each organization retains its independence but has the capability to communicate and interoperate with other organisations. Building upon this, the reference research projects on EI, INTEROP and ATHENA, have enlarged this definition to a holistic perspective where EI can only be achieved when addressed covering all levels of an enterprise (business, knowledge, applications and data) [28].

In some cases (e.g. collaborative product development), SE is a particular case of EI since it requires integration of all the disciplines and areas of expertise from different enterprises into a team effort forming a structured development process that proceeds from concept to production to operation and a latter disposal. It also considers both the business and the technical requirements of all customers with the goal of providing a quality product that meets all user demands [9]. Therefore, MBSE needs EI to describe and integrate the LC of a product or system.

However, besides the challenge of integration models from different disciplines, achieving that inside heterogeneous networks is still an ongoing challenge hindered by the fact that they are, intrinsically, composed by many distributed hardware platforms, software tools and ICT systems [29]. Such engineering models could be defined in different languages or semantics, thus morphisms are needed to describe and formalise the relations between them, and sustainability methodologies are needed to cope with market dynamics: manufacturing systems are constantly adapting to new market and customer requirements, thus answering the need to respond with faster and better quality production; new organizations are constantly entering and leaving networks, leading to a constant fluctuation and evolution of system models. All these factors are making interoperability difficult to maintain [7]. Due to this fact, the authors propose the MIRAI (Monitoring morphIsms to suppoRt sustAinable Interoperability of enterprise systems) framework to monitor the systems' interoperability through the morphisms previously defined and stored on a company CM. MIRAI detects changes in the interoperable environment, proposing the user morphism re-adaptations in the advent of harmonization breaking.

4.1 MIRAI Framework and Workflow

As illustrated in Fig.4, MIRAI has the objective of monitoring the existing mappings and model versioning's stored in each enterprise's CM and timely detect the changes in the morphisms, proposing to the user a possible solution and preventing a significant transient period where interoperability in the network is not assured. The detection is carried as soon as CM changes, triggering an agent to search model differences. Indeed, when a versioning on one of the MBSE models is detected, MIRAI triggers a warning and automatically proposes a new mapping morphims to the user. This new suggestion is based on the 5-tupple mapping expression proposed by the authors in [4] and it is used to describe the relationship among the different models that are used in the MBSE during the LC stages, regardless of the language used. Within this framework, a mapping is created to respond to the evolution, where according to the tuple MapT: ID is the identifier; MElems are the two mapped entities which can be the same as before or involve others; KMType is the same as before since for example a mapping at conceptual level cannot be changed to a instance level; the MatchClass is the new mismatch class (see Fig.5); and finally the Exp is evolved as well according to that MatchClass. After that, since the user might accept the proposal or not, the authors decided to endow MIRAI a learning process based on weights to help in the choice of MatchClass for the new mapping, and increasing intelligence over time.

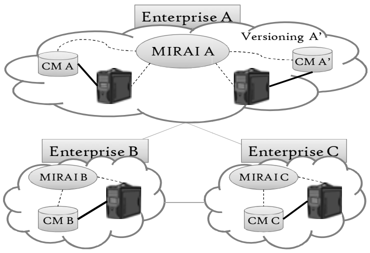


Fig. 4. MIRAI Network

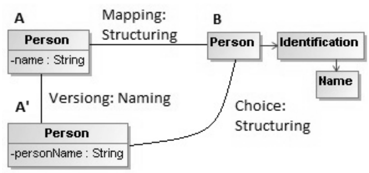


Fig. 5. Example of new MatchClass creation

Describing Fig. 5 in detail to illustrate a scenario, let's assume that two enterprises (A and B) are exchanging "Person" data to form design teams for a new product. In that case, a mapping morphism relates both structures. However by some reason, A evolved the way it models "Person" (versioning morphism) to A', and at that point the pre-existing mapping A →B needs to be revaluated since it might no longer be valid. MIRAI detects the evolution and proposes a new mapping, relating A' with B.

There are various types of mappings, which MIRAI could propose (e.g. Structuring, Semantics, Conceptual or Instantiable Data) to relate such model elements and depending on the user choice, the proposal is accepted and is stored in the CM. The next time a similar situation occurs, MIRAI will provide a similar solution only if it (from the various possible mappings types) remains being the most weighted one according to user's choice pattern. For reaching such objectives, MIRAI is directly associated to each CM. Moreover, within the collaborative network of enterprises there will be a kind of sub-network, i.e. the MIRAI network (as in Fig.4) that enables to keep all CMs syncronyzed (interoperable) and maximizes the learning process as the whole distributed framwork contributes with knowledge concerning user's selections, which is shared among the mediators.

4.2 Agents Architecture

MIRAI has been defined as a Multi-Agent System (MAS) framework due to the multiplicity of actions required, interacting with each other both indirectly (by acting on the environment) or directly (via communication and negotiation). As defined in literature [30], [31], an agent is *autonomous*, because it is able to work without action of the human, and has control over its actions and internal state; is *social*, because it cooperates with humans or agents, proposing new morphisms and learning with the user's choices; it's *reactive*,

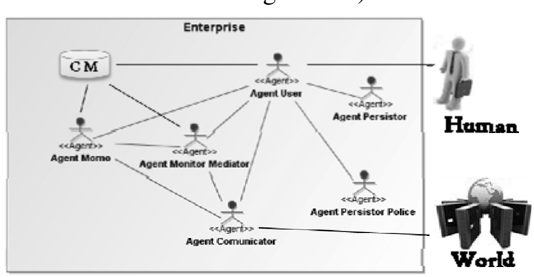


Fig. 6. MIRAI MAS Architecture

because it's perceives the environment and responds to the changes in the CM; and at last, the agent is *pro-active*, because it's not only reacting to the environment, but is able to take the initiative. Because of this and due to being a widely used agent-oriented middleware, JADE was used.

As illustrated in Fig. 6, six agents have been defined in the scope of MIRAI. Due to space constraints, only a highlight of their behaviour is here presented instead of their detailed formal workflows:

- *Agent Monitor Mediator* detects harmonization breaks by performing a scan in the CM in search for versioning's. Every time it finds a new morphims of this type it warns *Agent Momo* via the FIPA Query Interaction Protocol.
- *Agent Momo* acts when requested and makes the decisions in MIRAI. It checks if the changes occurred have an impact in the system's interoperable status, and if it does, contacts the *Agent User* proposing a possible solution (i.e. a new morphims or a correction to the existing one) according to *MapT* format of equation 1.
- *Agent User* is the interface between the MAS and the human user. Through this agent the human is informed of the solution proposed by the *Agent Momo* to decide whether if he/she accept it, or propose a new one;
- *Agent Persistor* controls if any agents dies, to resurrect it if required. This is done with the help of JADE, namely the Agent Management System (AMS). It receives the registration of all agents while managing their life cycle;
- *Agent Persistor Police* has a similar role but only controls the *Agent Persistor*, thus adding some redundancy to the MIRAI;
- *Agent Comunicator*, is responsible for the communication between different MIRAI's of a collaborative network. Every time an evolution happens and it is stored in the CM, this agent sends a warning to the other enterprises in the network informing what happened, enabling them to react as well. These communications are made using web services provided by JADE.

5 Case Study

In a typical manufacturing environment, when a new product is to be created, e.g. a novel car model, teams from different disciplines from design to aerodynamics, simulation, etc. need to cooperate, describing and modelling the several stages of the development LC of the product. In order to achieve this objective, some enterprises use MBSE strategies to help in that development and attain seamless integration between engineering processes. However, each enterprise has its own workflows and answers to several customer requirements, thus sometimes an interoperability disruption can occur, which complicates collaboration and data exchange.

The case study here presented describes a simplified collaborative network between three enterprises (A, B and C) that have collaborated in the past in the design of a sports car. They applied the MBSE paradigm, and in addition some have even worked with different modelling languages for structural models. After storing the mappings in the corresponding CM's following the MapT description of equation 1, each enterprise end ups with two high level morphisms (CMa: A→B and A→C; CMb: B→A and B→C; CMc: C→A and C→B) and the corresponding sub-morphisms to each of the relating entities and attributes. Relating to the previously presented Fig. 4, it is also possible to see MIRAI in each enterprise. Moreover, the figure also describes an evolution in the models of A to A', which after detection by Agent Monitor Mediator triggers the full system. Such an evolution may be caused by a new customer requirement, e.g. "build an eco-friendly sports car", and needs to be handled swiftly with the risk of jeopardizing the full network (see Fig. 7 to follow the A→A' and A→B mappings storyline). "Mapping 1" represents a part of the initial morphism A→B, and two mismatches are represented: 1) One (*Sub-Class Attribute*) is due to the enumeration attribute "combustion" from A being represented in B by a subclass hierarchy; 2) The other (*Coverage*) refers to the same attribute, since it can be lossy when looking from a bidirectional perspective, i.e. A is a subset of B.

On the top left side it is possible to see how the requirements model is linked with some parts of the structural model of A, namely the capacity that the last model has to answer the specific requirements.

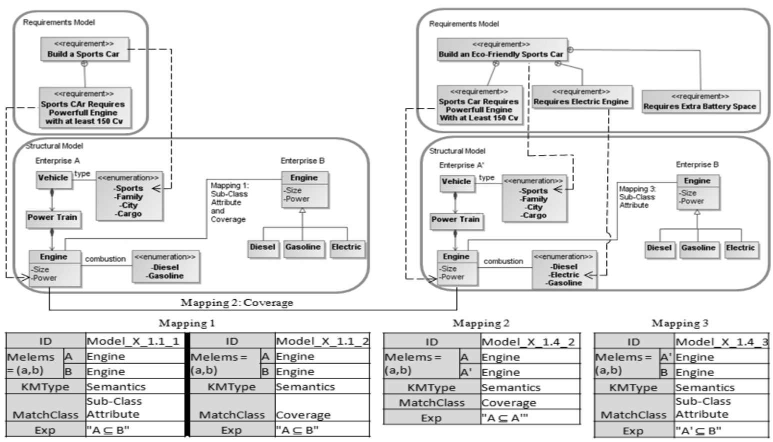


Fig. 7. Mappings Evolution

"Mapping 2" represents the evolution of the structural model of the engine from enterprise A (morphism A→A'), thus responding to the new requirement to be eco-friendly. Having an electric engine is a sub-requirement that the previous version of the model could not handle. As soon as this update is stored in CMa, MIRAI reacts and proposes a "Mapping 3" to re-establish full cooperation between A and B (morphism A'→B). In this case, the morphim with the *Sub-Class Attribute* mismatch is maintained, while the one with the *Coverage* mismatch is removed since now both A and B can handle the same sets of information. With this example it is possible to see how the MIRAI reacts when some modification in the CM occurs, and the advantages that MIRAI brings when used to monitoring models in SE relations.

6 Conclusions and Future Work

The practice of systems engineering has been transitioning from a document-based approach to a model-based approach like many other engineering disciplines. MBSE offers significant potential benefits to specification and design precision, it helps to improve design quality, productivity, and reduce development risk. The emphasis in MBSE is on producing and controlling coherent system models, and using them to specify a complete product throughout its lifecycle [15].

In the last years some languages for modelling systems that support MBSE have been developed. These support modelling of requirements, structure, behaviour and parametrics to provide a robust description of a system [14]. Until now, the best way to deal with interoperability depended on the usage of dedicated knowledge models and international standards acting as information regulators [7]. Yet, for different reasons some enterprises remain using traditional models for engineering analysis and simulation models, where each one tends to use its own data format and business rules, and handles as many mappings as the number software tools it needs to execute. This brings some issues in interoperability, becoming difficult to handle existing mappings, even when following MBSE paradigm.

The role of MIRAI is to monitor all the mappings that exist among the several models used by business partners in the same collaborative network, controlling the changes, warning and proposing new mappings, thus preventing interoperability problems that could cause a destabilization of the network harmony. Enterprises' privacy is assured since each one has its own MIRAI associated to an internal CM that tracks the morphims it maintains with is direct partners.

Such solution facilitates the creation of a network sustainable interoperability. This means that the systems are self-organized and capable of responding to environment changes, and network evidence system of systems behaviour. Systems' communication with the environment allows all necessary exchanges through its own boundaries, even though transformation mechanisms are influenced by internal rules, values, beliefs, constraints, culture, and internal models. This dynamicity is essential to support and sustain the interoperability applied to global business networks.

As future work, the authors intend to enhance the proposed framework with functionalities able to manage the dynamic inclusion or exclusion of enterprises in the network, together with an automatic detection of evolutions even if they are not updated in the CM knowledge base. Also, the agent *Momo* has space for improvement, enabling it with capabilities to analyse the network transients and provide more knowledgeable response.

References

[1] Hassell, L.a., Fogler, M.W., Russell, S.E.: Collaborative networks: helping rural laboratories achieve quality. Journal for Healthcare Quality: Official Publication of the National Association for Healthcare Quality 28, 44–49

[2] Schrage, M.: Shared Minds, p. 140. Random House, New York (1990)

[3] Chestnut, H.: Systems Engineering Tools. Wiley (1965) ISBN 0471154482

[4] Institute of Electrical and Electronics Engineers: IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, New York, NY (1990)

[5] Agostinho, C., Sarraipa, J., Goncalves, D., Jardim-Goncalves, R.: Tuple-based semantic and structural mapping for a sustainable interoperability. In: Proceedings of DoCEIS 2011 (2011)

[6] Haskins, C.: Systems engineering handbook. INCOSE. Version 3 (2006)

[7] Agostinho, C., Jardim-Goncalves, R.: Dynamic Business Networks: A Headache for Sustainable Systems Interoperability. In: 4th Int. Workshop on Enterprise Integration, Interoperability and Networking (EI2N 2009) of the OTM Conference (2009)

[8] Crisp II, H.: Systems Engineering Vision 2020. INCOSE. Version 2.03 (2007)
[9] INCOSE, http://www.incose.org
[10] Frisch, H.P., Stirk, C., Price, D., Spiby, P.: ISO STEP-AP233 Transition Development to Enablement. AP233 System Engineering (2007), http://www.ap233.org
[11] Nallon, J.: An Overview of AP-233: The Systems Engineering Data Exchange Standard based on STEP (ISO-10303). AP233 System Engineering (2003), http://www.ap233.org
[12] Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. In: ICSE 2000 Proceedings of the Conference on The Future of Software Engineering, New York (2000)
[13] Osvalds, G.: Using Simulation and Visualization to Support MBSE. INCOSE (2011)
[14] Friedenthal, S., Moore, A., Steiner, R.: A Practical Guide To SysML: The Systems Modeling Language. Morgan Kaufmann Publishers, Burlington (2008)
[15] Ogren, I.: On principles for model-based systems engineering. Systems Engineering Journal 3, 38–49 (2000)
[16] ISO/DIS 10303-233, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=55257
[17] ISO TC184/SC4: Product data representation and exchange—Part 11: Description methods: The EXPRESS language reference manual (2004)
[18] Modeling and Simulation Interoperability (MSI) Challenge Team, http://www.omgwiki.org/MBSE/doku.php?id=mbse:modsim
[19] Jardim-Goncalves, R., Agostinho, C., Malo, P.: Harmonising technologies in conceptual models representation. Int. J. Product Lifcycle Management 2(2), 187–205 (2007)
[20] Sarraipa, J., Beca, M., Agostinho, C., Jardim-goncalves, R.: Tuple-Based Morphisms For E-Procurement Solutions. In: Proceedings of the ASME 2011 International Design Engineering Technical Conference & Computers and Information in Engineering Conference IDETC/CIE 2011 (2011)
[21] Ehrig, M., Staab, S.: QOM - Quick Ontology Mapping, pp. 1–28
[22] Jardim-Goncalves, R., Agostinho, C., Steiger-Garcao, A.: A reference model for sustainable interoperability in networked enterprises: towards the foundation of EI Science Base. Under revision for IJCIM (2011)
[23] INTEROP NoE. Deliverable MoMo.2 - TG MoMo Roadmap. InterOP (2005)
[24] Dauben, J.W.: Georg Cantor: His Mathematics and Philosophy of the Infinite. Harvard University Press, Boston (1979) ISBN 978-0-691-02447-9
[25] Bernstein, P.A.: Applying Model Management to Classical Meta Data Problems. In: First Biennial Conference on Innovative Data Systems Research, California (2003)
[26] Sarraipa, J., Jardim-Goncalves, R., Steiger-Garcao, A.: MENTOR: an enabler for interoperable intelligent systems. Int. J. of General Systems 39(5), 557–573 (2010)
[27] Sarraipa, J., Zouggar, N., Chen, D., Jardim-Goncalves, R.: Annotation for enterprise information management traceability. In: ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (2007)
[28] AHENA IP: Specification of Interoperability Framework and Profiles, Guidelines and Best Practices. ATHENA public Deliverable D.A4.2 (2007)
[29] Ray, S.R., Jones, A.T.: Manufacturing interoperability. Journal of Intelligent Manufacturing 17(6), 681–688 (2006)
[30] Wooldridge, N., Jennings, M.: Intelligent Agents: Theory and Practice. The Knowledge Engineering Review 10(2), 115–152 (1995)
[31] Bellifemine, D., Caire, F., Greenwood, G.: Developing Multi-Agent Systems with JADE. Wiley (2007)

The Role of Coordination Analysis in Software Integration Projects*

Nuno F. Rodrigues[1,2], Nuno Oliveira[2], and Luís S. Barbosa[2]

[1] DIGARC - Polytechnic Institute of Cávado and Ave, 4750-810 Barcelos, Portugal
[2] DI-CCTC - Universidade do Minho, 4710-057 Braga, Portugal

Abstract. What sort of component coordination strategies emerge in a software integration process? How can such strategies be discovered and further analysed? How close are they to the coordination component of the envisaged architectural model which was supposed to guide the integration process? This paper introduces a framework in which such questions can be discussed and illustrates its use by describing part of a real case-study. The approach is based on a methodology which enables semi-automatic discovery of coordination patterns from source code, combining generalized slicing techniques and graph manipulation.

1 Introduction

Integrating running software applications, usually referred in the literature as the *Enterprise Application Integration* (EAI) problem [5,3], is one of most challenging tasks in enterprise systems development and management. According to Forrester Research, more than 30% of all investments made in information technologies are spent in the linkage of software systems in order to accomplish global coherent enterprise software solutions. Actually, tuning to new markets, fusion or acquisition of companies, evolution of legacy software, are just but examples of typical scenarios which entail the need for integration.

EAI aims at the smooth composition of services, data and functionality from different software systems, to achieve a single, integrated and coherent enterprise solution. Conceptually, however, a main issue behind most EAI projects concerns the definition and implementation of a specific *coordination model* between the systems being integrated. Such a model is supposed to capture system's behaviour with respect to its network of interactions. Its role is fundamental to help the software architect to answer questions like, which sub-systems are connected, how do they communicate and under which discipline, what are the dependencies between such connections, how do component's (w.r.t to integrating software systems) local constraints scale up to integrated systems, among many others.

* This research was partially supported by FCT in the context of the MONDRIAN project, under contract `PTDC/EIA-CCO/108302/2008`.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 83–92, 2011.
© Springer-Verlag Berlin Heidelberg 2011

If some sort of coordination strategy is a necessary component of any EAI project, what gets implemented on completion of the integration project often deviates from the envisaged strategy in a significant way. Reconstructing the implemented strategy and rendering it into a suitable model becomes, therefore, an important issue in EAI. On the one hand, such a reconstructed model plays a role in validating the (coordination component) of the integration process. On the other, it provides feedback to the software architect, eventually suggesting alternative strategies.

Such is the problem addressed in this paper. In a series of recent papers the authors have developed both a methodology [8,10] and a tool, CoordInspector [9] to extract, from source code, the structure of interactions among the different, reasonably independent loci of computation from which a system is composed of. The target of this methodology is what will be referred to in the sequel as the *coordination layer*, i.e. the architectural layer which captures system's behaviour with respect to its network of interactions[1]. The extraction methodology combines suitable slicing techniques over a family of *dependence graphs* built directly from source code, in the tradition of *program dependence graphs* (see, for example, [1,4]), which abstract all the relevant code information for interaction control. Such graphs are called *coordination dependence graphs* (CDG) in [10], where further details on their construction may be found. The tool processes CIL code, for which every .Net compliant language compiles to, thus making it potentially able to analyse systems developed in (combinations of) more than 40 programming languages.

The paper introduces a case study on verification of design time integration strategies. Section 2 sums up the methodology, and examples of its application are discussed in 3. Conclusions and pointers for future work are enumerated in section 4.

2 The Method

2.1 Coordination Patterns

Throughout this paper we adopt a *coordination-driven view* of software architecture in general and architectural analysis, in particular. On the other hand, *patterns*, in the context of this research, are placed at a low-level: they aim to be suitable representations of equivalence classes of (sub-graphs of) CDG extracted from code. Qualifier 'low-level' means that our focus is not on the description of traditional architectural styles, or even typical architectural elements, such as components, software buses or connectors, but the specification of architectural abstractions over dependence graphs extracted from code.

As an example consider the pattern depicted in Fig. 1 used to identify, in the client side of a service interaction, the so-called *asynchronous query pattern*

[1] The qualifier is borrowed from research on *coordination* models and languages [2], which emerged in the nineties to exploit the full potential of parallel systems, concurrency and cooperation of heterogeneous, loosely-coupled components.

with client multithreading. It is described in the graphical notation associated to CoordL, a domain-specific language for coordination patterns introduced by the authors in [6]. The corresponding textual version is shown in the right in a window of CoordInspector. Even if it is not the aim of this paper to provide a full description of CoordL, its graphical notation is almost self-explicative. For the moment, and to get a flavour of what coordination patterns are, note in this example how it encodes the following protocol: a client orders the execution of an operation in one thread, x, and then launches a second thread, y, to retrieve the result. Instances of this pattern are common whenever time consuming services are to be invoked and calling threads can not suspend until a response is returned.

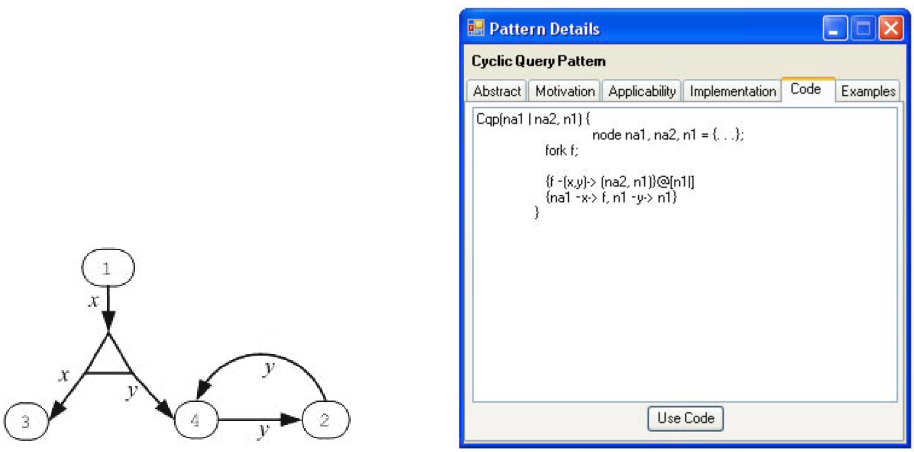


Fig. 1. Asynchronous query pattern with client multithreading

2.2 Discovering Coordination Patterns

Communication primitives, understood in the broad sense of any possible mechanism a component resorts to interact with another one, are the building blocks of the coordination layer of a software system. Direct foreign calls to well referenced components such as web-services calls, RMI or .Net Remoting calls to distributed objects are typical examples but, by no means, the only ones. The specific combinations of such primitives is what allows systems to control and interact, in complex ways, with other systems, processes, databases and services in order to achieve common goals. Thus, it is reasonable to expect that any coordination discovery strategy should start by identifying such primitive communication statements in the source code, together with the specific program context in which they are embedded. Therefore, our approach is *parametric* on the communication primitives as well as on the mode they are invoked (synchronous or asynchronous).

The reverse engineering process starts by the extraction of a comprehensive dependence graph from source code, the *Managed System Dependence Graph* (MSDG), which captures program statements in the vertices while the edges represent data, control and inter-thread dependencies. Then MSDG vertices containing primitive communication calls in their statements are singled out. We call this operation the *labelling* phase which is parametric on both the communication primitives and the calling mode. The result of this phase is another graph structure, the CDG, retaining only coordination relevant data with respect to the set of rules specifying the communication primitives to look for. The CDG is computed from the MSDG in a two stage process. First, nodes matching rules encoding the use of specific interaction or control primitives are suitably labelled. Then, by backward slicing, the MSDG is pruned of all sub-graphs found irrelevant for the reconstruction of the program coordination layer. Once the CDG has been generated, it is used to search for specific coordination patterns and trace them back to source code.

2.3 The Strategy

Recall the questions proposed in the introduction to this paper: *What sort of coordination strategies emerge in a software integration process? How close they are to the coordination component of the envisaged architectural model which was supposed to guide the integration process?*. This paper focus in *post-integration analysis*; thus on a verification of the coordination models developed before the integration process against the final, integrated system. Often, however, such models are only informally recorded, and relevant information scattered among documents describing the integration architecture.

The envisaged strategy has 4 stages. First the basic coordination solutions which were designed for the integration project have to be identified, by analysing the relevant documentation and, often, by interviewing the development team. In each case, a correspondent *coordination pattern* is specified in the diagrammatical notation of CoordInspector. At a third stage, such patterns are looked for in the source code of the integrated system, with the support of CoordInspector.

Often, however, patterns are discovered only in an incremental way. The strategy is to start the search with the pattern as described by the development team and, if it not directly found, split it into small patterns until a match is found. Then work on the reverse direction, re-building the shape of the patterns which is actually implemented. In the limit, the graph pattern is reduced to a unstructured collection of nodes (corresponding, e.g. to web-service calls) which the architect has to aggregate in patterns, in an iterative way. Actually, often what has been recorded, or what developers report, as the envisaged coordination policy is far from what is effectively implemented. Our approach helps in identifying and documenting such cases and also, once they have been retrieved, to discuss possible alternatives, eventually leading to improvements in the implementation. Such critical analysis is the fourth stage of our strategy.

Technically, coordination patterns are defined in a specific notation [6]. For a brief explanation consider Fig. 2 . A pattern is a graph, where upwards triangles represent the spawning of a new thread, downwards triangles denote a thread join, vertices contain predicates (composed by regular expressions) and edges represent control flow dependencies, both to be verified against a CDG. Edges bare two different types of labels, one capturing the number of edges to be matched on the CDG control flow edges (as in Fig. 2, this label can take the value +, standing for one or more edges), and a second label containing a variable to be bound to the thread id being matched on the CDG. This last label type avoids having to impose an order on the outgoing (incoming) thread edges of an upwards (downwards) triangle, which facilitates the graphical layout of the patterns.

3 A Case-Study in Architectural Analysis

This section illustrates the use of CoordInspector and coordination patterns in a particular case-study in architectural analysis. To respect space limits the real case-study is merely glimpsed here and a small detail taken for illustrative purposes.

The case-study, however, was a big one, dealing with the need to re-engineering a set of independent, but enduring software systems to achieve an effective degree of integration. It involved a Portuguese professional training company, with facilities in six different locations (see [8] for a complete description).

Before the integration project, the company relied in four main software systems, to be referred in the sequel as the *base components*. They comprised an *Enterprise Resource Planner* (ERP), a *Customer Relationship Management* (CRM), a *Training Server* (TS), and a *Document Management System* (DMS). The decision to integrate all these systems was pushed by the necessity of introducing a Web Portal, for on-line selling of both training courses and networking devices. Thus, the final system included these four existing components, plus the Web Portal developed during the integration project.

All those components operated in complete isolation from each other. Thus, every exchange of information between them was performed manually. The absence of integration led to numerous information synchronisation problems which had to be dealt with manually, at a daily basis. A sudden growth in the company business level, made it no longer feasible to maintain all the information synchronised manually. Actually, several incoherencies in critical company data inevitably started to emerge. When the integration project was launched an administrative decision forced the choice of a point-to-point integration architecture.

Although the case-study encompassed the whole integration process we will concentrate our attention here on a specific problem related to consistent data update. The problem resided in the Web Portal application, the component which was laid responsible for the user's data update across all systems.

The re-engineering process started with an attempt to recover from the relevant components source code all the coordination protocols governing user's data update.

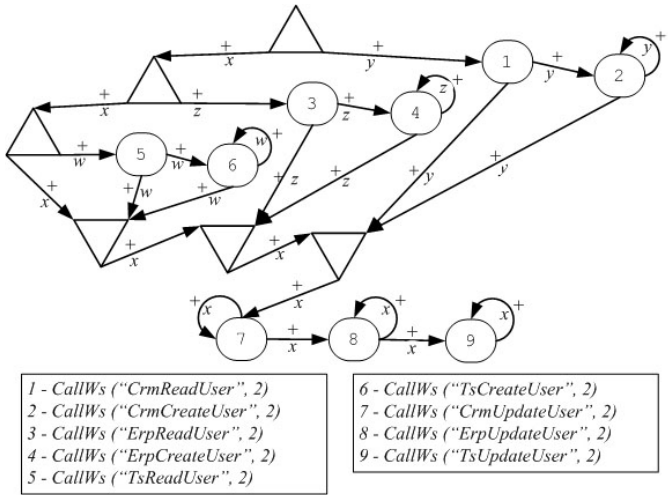


Fig. 2. User update operation

Fig. 2 depicts a common coordination pattern for updating user's data across applications, found recurrently in the system. This patterns however is not able to prevent the creation of already registered users (which in the generic description of the pattern would be translated to recurrent invocations of nodes 2, 4 and 6). To prevent this, the pattern can be changed by inserting extra vertices for checking if a user already exists in the relevant data base component. However, the calls to the remote creation operation (nodes 2, 4 and 6) are always carried after a read operation (nodes 1, 3 and 5), which forces the first remote call to the former to be aware of an eventual previous registration of the user. Thus, only the subsequent remote creation calls (executed through loop edges $2 \rightarrow 2$, $4 \rightarrow 4$ and $6 \rightarrow 6$) suffer from the problem of inserting duplicate users. Therefore, a small change is enough: insert user existence check nodes (nodes 3, 6 and 9 in Fig. 3) after each remote creation call.

The remote update of a user is only performed at the very end of this coordination pattern, in nodes 7, 8 and 9 (Fig. 2) or 10, 11 and 12 (Fig. 3). Moreover, all updates occur in a single thread. This opens the possibility of a previous call introducing delays in subsequent calls, resulting in significant delays for the overall remote operation. This single thread sequence of remote calls also demands for a rigorous exception and error handling, given that each call may influence subsequent ones and consequently the entire operation. In this case-study, once the pattern was discovered, manual inspection of error handling routines was required, because these mechanisms were, then, not incorporated in the CoordL pattern language.

Once identified, this pattern was improved by replacing the sequence of update calls by their parallel execution, as represented in Fig. 3. This potentially minimised the effect of delays introduced by individual calls. A possibility remains, however, for a remote call to continually fail. In such a case, this pattern

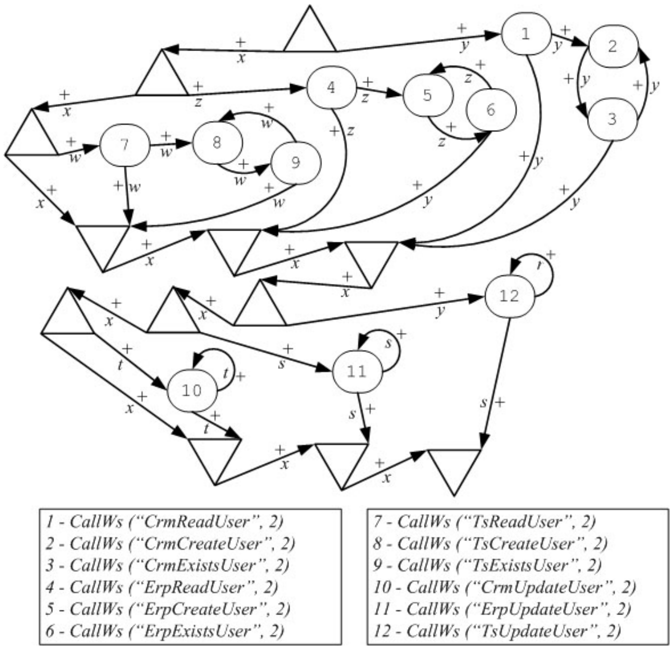


Fig. 3. Corrected user update operation

may not only fail, but, what is worse, deadlock, which could ultimately lead to a complete halt of the entire ECS system. Note that the discovery of such deadlock situations was made easily by using the discovery algorithm to look for loop patterns.

A solution to avoid deadlocks consists of introducing a counter for each identified loop and include a guard (or extend one, if a guard was already there) in the loop to inspect the number of cycles performed. In case one of these loops has overcome the maximum number of cycles allowed, the guard not only guarantees that the program control leaves the loop, but also that the operation not carried out is written to an *error log* table. The deadlock removal strategy introduces a mechanism for error recovery and enables the introduction of different amounts of tries for each remote call. Furthermore, the *error log* table can be used, for instance, to run periodically a *batch* job responsible for the re-invocation of failed operations.

What is important to underline at this stage is the method which lead to this new protocol. First the problem was clearly identified by recovering, from source code and at different components, the pattern corresponding to the protocol in use. Then CoordInspector was used again to identify the associated loops, source of possible deadlocks. Finally the knowledge gathered along the analysis process was used to design a new solution, encode it as a new coordination pattern and its integration back into the repository.

Another example of how effective this approach is is provided by coordination protocol for the online sale of a set of training courses. The pattern actually extracted with CoordInspector is depicted in Fig. 4. Without entering in detail, it is easy to recognise a purely sequential flow of activities.

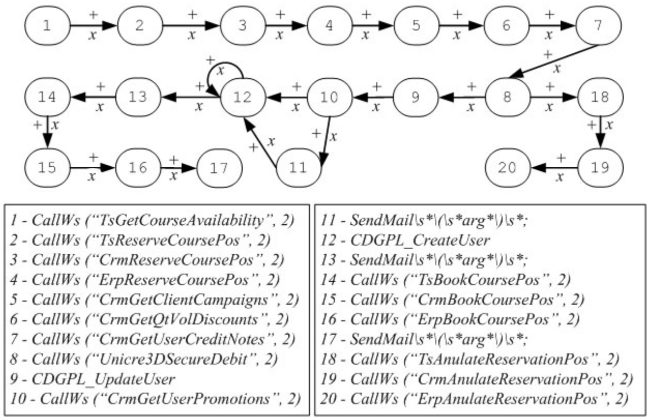


Fig. 4. Training courses sale operation

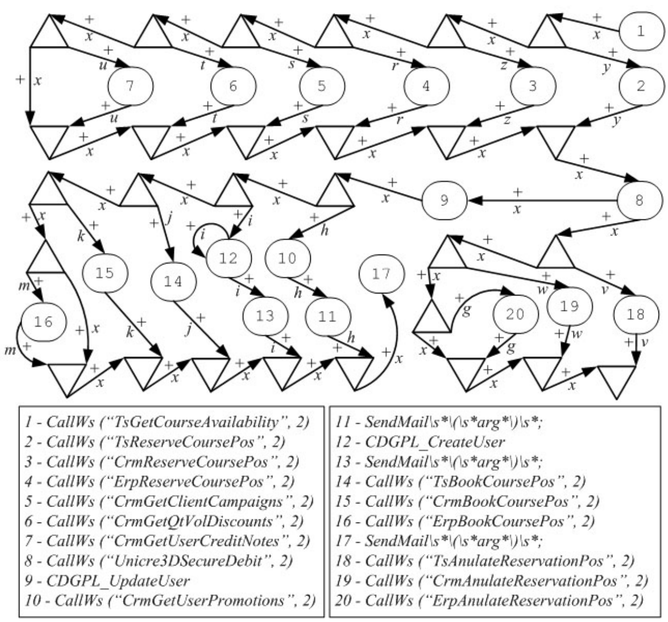


Fig. 5. Improved training courses sale operation

Note that, even though the user update and create operations are multithreaded, the actual sale operation is entirely performed on a single thread. This, of course, degrades performance, while it would be possible to execute in parallel several activities that do not depend on each other.

Therefore, the system's integrator has proposed the modification of this pattern as depicted in Fig. 5. The main modification concerns the introduction of concurrency between every independent activity. However, some activities still have to be performed in sequence. For example, vertex 9 is executed after vertex 8 because updating (in the former vertex) depends on the completion of the payment operation contained in the latter.

4 Conclusions and Future Work

The need for methods and tools to identify, extract and record the coordination layer of software systems is becoming more and more relevant as an increasing number of software systems rely on non trivial coordination logic for combining autonomous services, typically running on different platforms and owned by different organisations. Actually, if coordination policies can be extracted from source code and made explicit, it becomes easier to understand the system's emergent behaviour (which, by definition, is the behaviour which cannot be inferred directly from the individual components) as well as to verify the adequacy of the software architecture (and of the code itself) with respect to expected interaction patterns and constraints.

This seems particularly relevant in the context of software integration projects, as discussed in the paper. We proposed a tool-supported methodology for recovering, from source code, the coordination strategies *as-implemented* in a software integration project. The whole process is driven by the views of such strategies *as-stated* in the project documentation, often in an informal and vague way. The case study illustrates further how this sort of analysis can be useful in validating design decisions, taken during the integration stage, eventually improving the final system or, at least, providing a precise documentation of its (coordination) architecture.

In the case study reported here, we were able to detect and correct several coordination problems even before these have showed any evidences of themselves as runtime errors or data inconsistencies between the integrated components. Moreover, a number of subtle, yet important, coordination mistakes (e.g., the wrong order of execution of two statements) were detected, that would be much more difficult to discover by manual code inspection.

To the best of our knowledge, this line of enquiry in reverse architectural analysis, focused on coordination issues, is largely unexplored. There is, however, a lot of work on automatic gathering and registering of architectural information understood in a more classical way as the *system's gross structure*. Among others, the Alborz system [11] and the DiscoTect platform [12] should be mentioned. Alborz presents the architecture as a graph of components and keeps a relationship between this graph and the actual source code. The same applies to Bauhaus [7], a system which maps architectures to graphs whose nodes may represent types, routines, files or components and the edges model relationships between them, from which different architectural views can be generated. A detailed comparison with our own, largely complementary work is still lacking. Future developments of this work include a number of improvements to CoordInspector, namely the inclusion of the possibility of directly modifying the implementation code by editing the sub-graphs identified in the analysis process and a performance study concerning different integration scenarios.

References

1. Ferrante, J., Ottenstein, K.J., Warren, J.D.: The program dependence graph and its use in optimization. ACM Trans. Program. Lang. Syst. 9(3), 319–349 (1987)
2. Gelernter, D., Carrier, N.: Coordination languages and their significance. Communication of the ACM 2(35), 97–107 (1992)
3. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Longman Publishing Co., Inc., Boston (2003)
4. Horwitz, S., Reps, T., Binkley, D.: Interprocedural slicing using dependence graphs. In: PLDI 1988: Proceedings of the ACM SIGPLAN 1988 Conf. on Programming Usage, Design and Implementation, pp. 35–46. ACM Press (1988)
5. Linthicum, D.S.: Enterprise application integration. Addison-Wesley Longman Ltd., Essex (2000)
6. Oliveira, N., Rodrigues, N., Henriques, P.R., Barbosa, L.S.: A pattern language for architectural analysis. In: 14th Brazilian Symposium in Programming Languages, SBLP 2010, Slavador, Brasil, vol. 2, pp. 167–180. SBC — Brazilian Computer Society (September 2010) ISSN: 2175-5922
7. Raza, A., Vogel, G., Plödereder, E.: Bauhaus - a Tool Suite for Program Analysis and Reverse Engineering. In: Pinho, L.M., González Harbour, M. (eds.) Ada-Europe 2006. LNCS, vol. 4006, pp. 71–82. Springer, Heidelberg (2006)
8. Rodrigues, N.F.: Slicing Techniques Applied to Architectural Analysis of Legacy Software. PhD thesis, Escola de Engenharia, Braga, Portugal (2008)
9. Rodrigues, N.F., Barbosa, L.S.: Coordinspector: a tool for extracting coordination data from legacy code. In: SCAM 2008: Proc. of the Eighth IEEE Inter. Working Conference on Source Code Analysis and Manipulation, pp. 265–266. IEEE Computer Society (2008)
10. Rodrigues, N.F., Barbosa, L.S.: Slicing for architectural analysis. Science of Computer Programming (March 2010)
11. Sartipi, K., Dezhkam, N., Safyallah, H.: An orchestrated multi-view software architecture reconstruction environment. In: 13th Working Conference on Reverse Engineering (WCRE 2006), Benevento, Italy, October 23-27, pp. 61–70 (2006)
12. Schmerl, B.R., Aldrich, J., Garlan, D., Kazman, R., Yan, H.: Discovering architectures from running systems. IEEE Trans. Software Eng. 32(7), 454–466 (2006)

Information Centric Frameworks for Micro Assembly

Raviteja Gunda[1], Joe Cecil[1], Prasad Calyam[2], and Subhash Kak[1]

[1] Oklahoma State University, Stillwater, Oklahoma, USA
{raviteja.gunda,j.cecil,subhash.kak}@okstate.edu
[2] Ohio State University, Columbus, Ohio, USA
calyam.1@osu.edu

Abstract. This paper outlines an information centric framework involving cyber architectures and virtual reality simulation resources for micro devices assembly. Such a framework lends itself to cyber physical paradigms which hold significant potential in todays distributed collaborative engineering environments. The emerging domain of micro assembly is used to highlight the architecture and overall approach. An overview of the cyber architecture, and engineering modules to support planning, analysis, and simulation of target micro assembly activities is provided.

1 Introduction

When a micron sized part design has complex shapes and material properties, it is difficult to use Micro Electro Mechanical Systems (MEMS) technologies. In such situations, there is a need to use micro assembly technologies. Micro Devices assembly refers to the assembly of the micron sized parts using manual, semi automated or automated techniques (a micron is 10-6 meters). Typically, robotic or other assembly resources are used to develop such work cells that can rapidly assembly such complex and miniaturized components.

Micro devices assembly is an emerging domain with a significant economic potential. This technology has the potential to be used to develop biomedical devices, sophisticated miniaturized electronic sensors and surveillance devices as well as build more complex chemlab-on-a-chip type of products.

In this paper, our emphasis is on the design of an information centric framework which will accomplish the following:

1. Enable distributed resources involved in the life cycle of a micro assembly enterprise to collaborate
2. Facilitate virtual reality based simulation of the target assembly processes.

The goal of such a framework is to enable the rapid assembly of micro devices using distributed resources integrated through networks such as the Internet.

2 Literature Review

Our earlier research in creation of virtual environments can be found in [1-4]. In [1, 3] the development of a VR based approach to manipulate and assemble micron sized

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 93–101, 2011.

© Springer-Verlag Berlin Heidelberg 2011

parts is outlined. In this approach, the candidate's assembly plans are virtually studied before the assembly instructions are downloaded to a micro assembly cell. Genetic algorithms based assembly sequencing is used to generate assembly sequences [3]. This work cell used a gripper from Zyvex Corporation. Two versions were created of the Virtual Reality environments (one on Unix platform and another on the PC platform). A more exhaustive review of research in micro assembly can be found in [2, 4, 8]. In [4], an advanced work cell is outlined to assemble micron sized parts.

A brief note on MDA is important to highlight this emerging process domain of importance. Micro devices assembly and Micro Electro Mechanical Systems (MEMS) are two distinctly different but related areas. When part designs are complex in shape and require the use of material of varying properties, they cannot be manufactured using MEMS technology but have to be 'assembled'; this field, which involves on the assembly of micron sized parts, can be described as 'Micro Devices Assembly' (MDA). Manual assembly of micron-sized parts is difficult and time consuming. Innovative computer based automated assembly methods must be developed to increase efficiency, reliability, and reduce cost. Micro devices can be used in imaging applications, in medical surgery, etc. Research initiatives are underway at a variety of universities and government labs to address a range of issues from design of grippers to creation of VR based environments to assist in the rapid assembly of micro devices [8].

In [5], a hybrid approach to the assembly of micro sized components is proposed. This approach involves use of VR based simulation technology along with machine vision based servoing techniques.

The specific assembly routines are first planned in the VR environment and subsequently commands are sent to the physical work cell where the corresponding assembling tasks are completed. Camera based vision systems provide feedback which is used to make further adjustments.

In [6], a teleoperated micro assembly work cell is outlined that integrates a VRML-based virtual reality with visual servoing micro manipulation strategies. Teleoperation and visual servoing strategies are the key aspects of this paper. In [7], Probst et al. outlined the integration of a virtual reality system into an existing physical world for micro assembly activities. An advanced micro assembly system with six degrees of freedom, three cameras with a microscope and an illumination system is discussed. The virtual reality environment is a basic system which enables users to view the work cell in different angles and helps accomplish micro assembly operations without collisions.

In [9], the authors have outlined a real-time 3D visual tracking method and the 3D vision-based control approach. A model-based tracking algorithm is used to send 3D poses of the MEMS to ensure precise control. In [10], a hybrid micro assembly system with sensory feedbacks is proposed that uses both the vision-based microassembly and the scaled teleoperated microassembly. It has 6 DOF micromanipulation units, a stereomicroscope, and a haptic interface for force feedback which ensures high precision and dexterity in micro assembly.

In [11], a hybrid universally handling systems with an assembly head, six-axis fine positioning system, and an integrated sensor unit for microassembly is discussed. In [12], an outline is provided of the factors affecting the increment of strain values and optimal location of strain gauges for force sensing.

3 Cyber Architecture of Proposed Approach

There are unique reasons for the creation of a collaborative distributed approach for the domain of MDA. A major problem worldwide in MDA is the lack of software and physical resources as it's an emerging and expensive domain. Various academic, government and research enterprises possess a diverse set of resources for assembling a range of micro devices. As Micro assembly resources are expensive and various partners typically possess different physical equipment and software resources, there is a need to form collaborative partnerships to address specific customer requests related to Micro Assembly. The physical MDA cells at location A (e.g.: New York) can be used to link with other software environments in Italy and China (for example); collectively, they can collaborate and use a diverse set of resources capable of planning, simulation, analyzing and assembling micron parts; collectively, this set of resources will can be viewed as a Virtual Enterprise (VE).

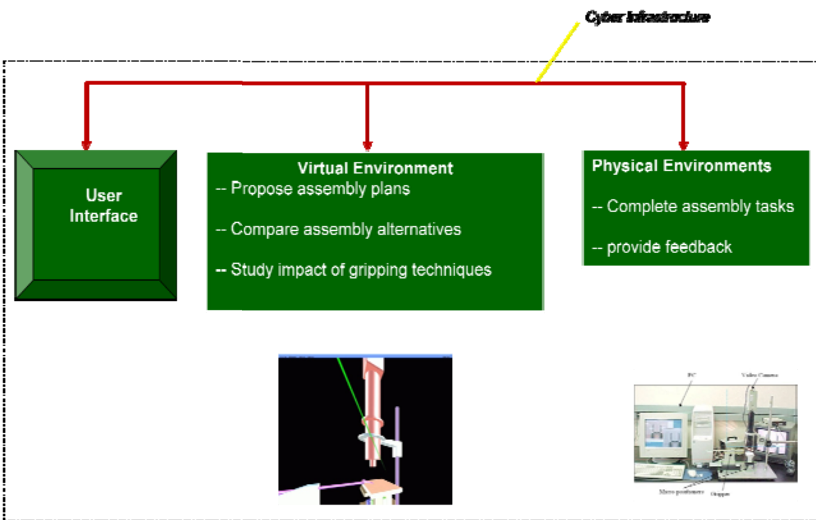


Fig. 1a. Information Centric Framework for Micro assembly

In this section, an outline of the proposed IC framework is discussed. The 3 main components include

a) User interface
b) Virtual Environment (to plan and simulate micro assembly tasks)
c) Physical Environment to implement target micro assembly tasks.

These and other resources are distributed and linked by computer networks such as the Internet (Internet 2). Using these resources, users can also propose assembly plans or generate assembly plans automatically using a genetic algorithm based approach. Using the VR environment, users can compare various assembly plans and select the most feasible. After the virtual analysis, a user can forward that assembly plan to the physical environment to complete the target assembly tasks. The physical environment can provide feedback to the user using cameras.

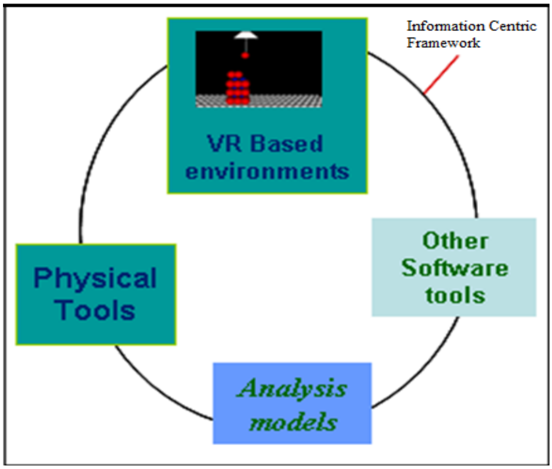


Fig. 1b. Distributed resources in the IC Framework

The other resources are necessary to accomplish the 'mini' life cycle of this distributed virtual enterprise (VE). An advanced IT architecture is necessary to enable this VE to respond to changing customer requirement. In our modeled context, a user can upload their micro assembly design through user interface.

Currently, initial development of this IC framework is underway. A VR Power Wall (from Mechdyne) is used to create the virtual environments. User can interact using Stereo Eye wear and a controller for visualizing the 3D simulations and for navigation. The controller can be used to navigate through the virtual environment and helps to select / modify the assembly plan. Motion trackers are used to track the user's position in the virtual environment.

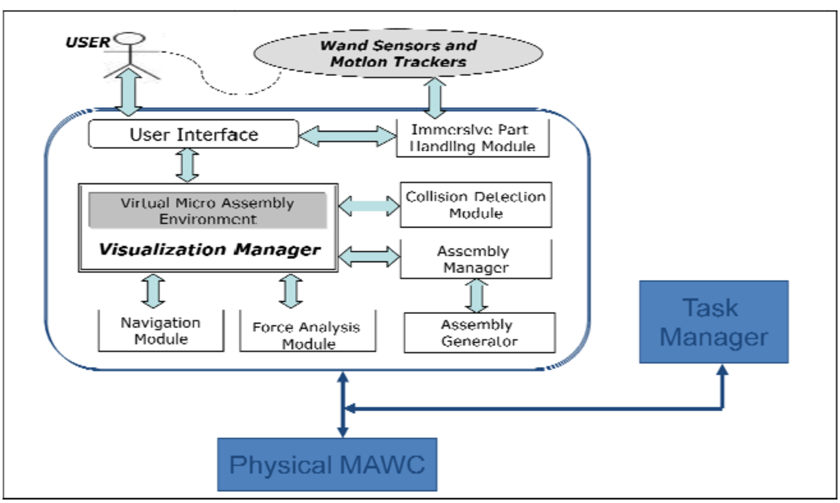


Fig. 2. Architecture of Virtual environment

An outline of the proposed modules follows:

a) *Navigation Module* allows the user to zoom in/out and explore the VR environment.
b) *Immersive part handling module* allows the user to pickup objects and make changes in the virtual environment.
c) *Collision Detection Module* is used to detect the collisions in the process of micro devices assembly. User can select the auto mode of collision avoidance assembly plans or user can make the changes to the assembly plan through user interface module.
d) *Assembly Manager* is responsible for the generation of the automatic assembly plans with the help of assembly generator or accepts manual inputs by the user.
e) *Force Analysis Module* is a planned module for the future where the impact of adhesive forces can be studied virtually (when comparing candidate gripping surfaces and techniques).
f) *The Task Manager* is responsible for overall completion of all cyber physical tasks.

4 Architecture of Work Cells 1 and 2

In this framework, a number of physical resources can be considered. In our setup, we have two work cells (see fig 3 and fig 5). They are referred as Work cell 1 and 2.

The original work cell 1 was reported in [3] which employed a gripper from Zyvex Corporation. Currently, the work cell 1 uses a gripper from FEMTO Tools. The assembly plate has 3 degrees of freedom in x, y, and z direction. (Micro positioner 1 moves left and right along x-axis; micro positioner 2 moves towards and away from the user along z-axis, and micro positioner 3 moves up and down along y-axis). The gripper is mounted at the end of a wooden rod, which is connected to a linear micro positioner, which moves longitudinally along the x-axis. An overhead camera is used to monitor and control the movement of these micro positioners and gripper. A virtual environment developed of work cell 1 allows the user to input the proposed assembly plans or generate automatic plans. Subsequently, the physical assembly tasks are implemented using the micro positioners, gripper and camera system. Figure 4 provides a view of this virtual environment. This was created using Coin 3D and C++.

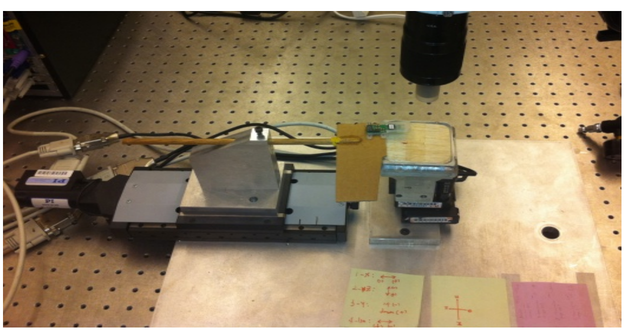


Fig. 3. Physical work cell 1

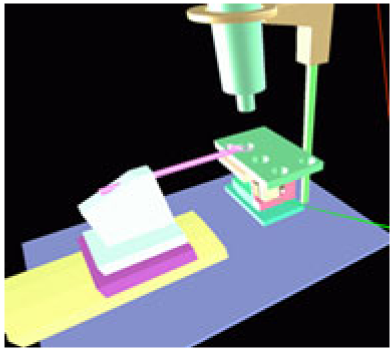


Fig. 4. Work cell 1 in virtual environment

4.1 Working of Work Cell 2

Work cell 2 has an advanced gripper which can assemble micro devices on an assembly work surface which has a rotational degree of freedom. The base of the assembly surface has 2 linear degrees of freedom (x and y axis); the gripper can move up/down along the z axis and can be manually adjusted for various tilt angles to enable better grasping of target parts (figure 5). Camera and illumination components are part of work cell 2. A variety of tweezers can be used as grippers to manipulate micron sized objects (the gripper is capable of being interchangeable). The virtual environment for work cell 2 is capable of allowing users to manually input path plans (figure 6). With the help of Coin 3D, Cavelib and C++, this environment can be run on a non immersive PC environment or on the immersive Power Wall in the Center for Information based Bioengineering and Manufacturing (CINBM). Using motion trackers and stereo eyewear, a user can navigate and propose alternate assembly plans using these environments. While one user's position is tracked, up to four other users can also be immersed in the same virtual environment simultaneously. This enables team based problem solving and analysis.

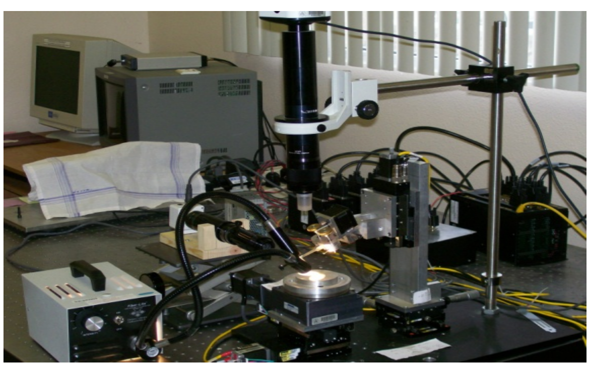


Fig. 5. Physical work cell 2

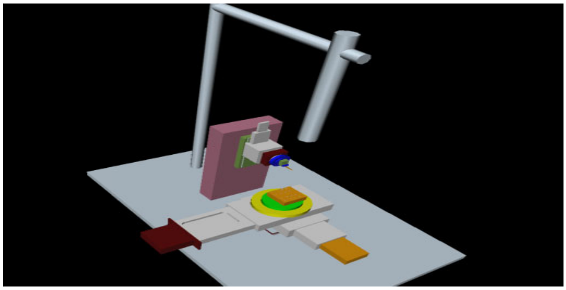


Fig. 6. Work cell 2 in virtual environment

5 Using Measurement Services in the Information Centric Framework

One of the innovations in our Information Centric Framework design is the exploration of cloud computing principles to share among various distributed locations the simulation graphics (corresponding to specific analysis using the VR environments) as well as real time videos of the micro assembly activities (accomplished by the work cells). In our approach, we are exploring the use of software tool called OnTime Measure. This measurement service can be used to perform centralized / distributed control and provisioning of measurements *within* experiment slices for various activities including the following:

- Network performance anomaly detection
- Network paths monitoring
- Network weather forecasting
- Network bottleneck fault location diagnosis.

OnTimeMeasure provides a research web portal that allows users to create and monitor experiments on geographically distributed systems. OnTimeMeasure-GENI is a measurement service for the GENI facility users, which is under development by researchers at Ohio State University's Supercomputing Center as part of the GENI initiative [14].

The Global Environment for Network Innovations (GENI) is a National Science Foundation (NSF) initiative which can be described as the creation of a unique virtual laboratory for designing and testing innovative ideas related to future Internets. The emphasis is on conducting at-scale networking experimentation where a diverse group of researchers can collaborate on the design of future internets. The overall mission of GENI mission is two-fold:

- Encourage and establish a basis for transformative research at the frontiers of network science and engineering; and
- Foster the potential for groundbreaking innovations of significant socio-economic impact [13].

By adopting use of OnTime as part of the IC framework for micro assembly, distributed teams and engineers can share the simulation views (or videos of physical tasks) very efficiently in run time. Today's internet provides low quality video sharing that provides access to a larger number of users; this, however, compromises the quality of the video received by the users who can be geographically distributed. OnTimeMeasure resolves this issue by maintaining measurement services time to time while improving the video quality based on the response from each user. It achieves this without compromising the quality of the video while providing effective access to the maximum possible users.

OnTimeMeasure deploys Node Beacons at each location in the distributed environment. It will provide this Information Centric Framework (ICF) the capability to fulfill the on-demand measurement requests. Node Beacon's tools can be installed on the ICF network which can provide up-to-date information on delays, bandwidth, data loss, jitter, and route changes etc., based on a schedule. With the information gathered from each distributed resource in the ICF, OnTimeMeasure improves the quality of the data transfer to each distributed location. Consider a scenario involving three computers at different locations with 'agents' residing at each site. Agents are the information collectors who gather specific information and send them to servers as well as retrieve the information as necessary. Data is shared between the sites with relevant control signals which can be used to start and stop the data sharing involved. In this scenario, the database located at each site stores the data associated with the simulation graphics (as they are executed on a specific computer at one location) as well as video monitoring of the final (physical) assembly of Micro devices by the appropriate work cells.

6 Life Cycle of Micro Devices Assembly

The main tasks in the life cycle of this context include

a) Create assembly plan
b) Perform simulation and assembly plans
c) Detect collisions and evaluate assembly plans
d) Make modifications
e) Identify gripper to be used
f) Implement assembly plan
g) Obtain feedback on completion of tasks using cameras.

Task manager supervises the physical and virtual tasks.

7 Conclusion

In this paper, we proposed an innovative IC framework for micro assembly. This IC framework has 2 major facets:

1. Enable the use of distributed resources involved in the life cycle of a micro assembly enterprise to collaborate
2. Facilitate virtual reality based simulation and analysis of the target assembly processes.

Such a frame work will enable the integration of distributed cyber physical resources for the field of micro assembly. There are unique reasons for the creation of a collaborative distributed approach for the domain of MDA. A major problem worldwide in the domain of MDA is the lack of adequate software and physical resources as it's an emerging and expensive domain. Various academic, government and research enterprises possess a diverse set of resources for assembling a range of micro devices. As Micro assembly resources are expensive and various partners typically possess different physical equipment and software resources, there is a need to form collaborative partnerships to address a varied set of customer requests related to Micro Assembly. The design of an Information Centric framework fosters such collaborations in a Virtual Enterprise context where a diverse group of industrial organizations work together to respond to customer's request.

References

[1] Cecil, J., Powell, D., Gobinath, N.: Micro devices assembly using Virtual environments. Journal of Intelligent Manufacturing 18(3), 361–369 (2007)
[2] Cecil, J., Vasquez, D., Powell, D.: Assembly and Manipulation of Micro Devices-A State of the Art Survey. Robotics and Computer-integrated Manufacturing 23(5), 580–588 (2006)
[3] Gobinath, N., Cecil, J.: Development of a Virtual and Physical Work Cell to assembly micro-devices. Robotics and Computer Integrated Manufacturing 21(4-5), 431–441 (2005)
[4] Boettner, S.S., Cecil, J., Jiao, Y.: An Advanced Collaborative Framework for Micro Assembly. Automation Science and Engineering, 806–811 (2007)
[5] Cassie, C., Ferreira, A., Hirai, S.: Combination of Vision Servoing Techniques and VR-based Simulation for Semi-Autonomous Microassembly Workstation. In: IEEE: International Conference on Robotics and Automation, pp. 1501–1506 (2002)
[6] Joseph, A., Barmeshwar, V., Bradley, J.N.: A Virtual Reality Teleoperator Interface for Assembly of Hybrid MEMS Prototypes. In: ASME Design Engineering Technical Conference (1998)
[7] Martin, P., Christoph, H., Ruedi, B., Bradley, J.N.: Virtual Reality for Microassembly. In: Institute of Robotics and Intelligent Systems, ETH Zurich, Switzerland (2007)
[8] Cecil, J.: Virtual Engineering, pp. 1–15. Momentum Press, LLC (2010)
[9] Dembele, S., Tamadazte, B., Be Fort-Piat, N., Marchand, E.: CAD Model-based Tracking and 3D Visual-based Control for MEMS Microassembly. The International Journal of Robotics Research 29(11), 1416–1434 (2010)
[10] Kim, D.-H., Park, J.-O., Kang, H., Kim, B.: A flexible microassembly system based on hybrid manipulation scheme for manufacturing photonics components. The International Journal of Advanced Manufacturing Technology 28(3), 379–386 (2006)
[11] Brecher, C., Wenzel, C., Freundt, M.: Hybrid Universal Handling Systems for Micro Component Assembly. Microsystem Technologies 14(12), 1855–1860 (2008)
[12] Lee, S.M., Choi, H.R., Yong Seok, C.: An Enhanced Force and Contact Position Sensor for Micro-Manipulations. International Journal of Control, Automation and Systems 7(3), 459–467 (2009)
[13] Global Environment for Network Innovations (GENI), http://www.geni.net/?page_id=2
[14] OnTimeMeasure, http://ontime.oar.net/

Towards Interoperability through Inter-enterprise Collaboration Architectures

Alix Vargas, Andrés Boza, and Llanos Cuenca

Centro de Investigación en Gestión e Ingeniería de Producción (CIGIP),
Universitat Politècnica de València,
Camino de Vera s/n Ed 8G -1º y 4º planta Acc D,
(Ciudad Politécnica de la Innovación)
alvarlo@etsii.upv.es,
{aboza,llcuenca}@cigip.upv.es

Abstract. Most enterprise architectures published so far are capable of generating reasonably good descriptive models for individual enterprises to enable integration, organization and synchronization of enterprise elements: organizational structure, business processes, information systems and technology infrastructure, among others. However, research in this field applied to the extended enterprise or inter-enterprise architectures that takes into account the growing trend towards complex collaborative environments is very scarce. In this sense, this article seeks to analyze, link and synthesize the researches that has addressed the disciplines of enterprise architecture and business collaboration, in order to identify possible future research needs from the conceptualization made.

Keywords: Collaborative networks, enterprise architectures, interoperability.

1 Introduction

In recent years, it has been demonstrated the interest that is taking the enterprise engineering through the use of enterprise architectures, evidenced by the large number of publications found about. However, these methodological proposals are mostly directed and oriented to individual enterprises. Therefore, the focus of these researches are not taking into account the growing and rapidly evolving business collaboration environments where two or more companies involved in one or more supply chains make collaborative supply networks. The operation and organization of these networks should be structured and modeled through collaboration architectures designed with the purpose of supporting collaborative processes and their integration with information systems of partners involved in this collaboration. In this sense, some frameworks have emerged seeking to propose a general framework for collaboration [1], [2], [3] and [4]. However, most of these studies focus their efforts on the proposed framework that in the procedure to ensure effective implementation. Another factor to highlight in these researches is that they propose different modeling

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 102–111, 2011.
© Springer-Verlag Berlin Heidelberg 2011

languages, which makes difficult interoperability between different network architectures.

The aim of this paper is to make a detailed analysis of the state of the art of these two disciplines both separately and jointly, in order to build the bases to make a proposal in a future work of an inter-enterprise collaboration architecture to ensure interoperability and synchronization processes belonging to the global domain of a collaborative supply chain.

2 Conceptualization of Enterprise Engineering and Enterprise Architectures

2.1 Enterprise Engineering

Enterprise Engineering is the discipline applied in carrying out any efforts to establish, modify, or reorganize any enterprise [5]. This discipline is responsible for defining, structuring, designing and implementing enterprise operations as communication networks of business processes, which comprise all their related business knowledge, operational information, resources and organization relations [6]. The field of Enterprise Engineering is concerned to understand, define, design and redesign business entities, which includes all knowledge, and organizational relationships, as well as life cycle [7]. Therefore, Enterprise Engineering facilitates the integration of all elements of the enterprise.

2.2 Enterprise Integration

As for the concept of enterprise integration, several authors provide their own definitions: [6; 7; 8; 9; 10; 11; 12]. In order to condense, integrate and collect the most important concepts of these definitions, we propose the following definition:

"Enterprise Integration is an enterprise approach that seeks to relate synergistically the elements of the company such as information, resources, applications and people, thus enabling the company to act as a total joint system which increases cooperation, coordination and communication between its components with the ultimate goal of fulfilling the mission and goals of the organization in a productive and efficient form".

2.3 Enterprise Architecture

Achieving Enterprise Integration through the Enterprise Engineering is possible thanks to the use of Enterprise Architectures. Conceptually, the enterprise architecture can be defined as:

"A discipline that provides a set of principles, methods, models and tools used for analysis, design and redesign of a company, thus allowing to represent and document the elements that form the company (such as organizational structure, business processes, systems information and technology infrastructure) and the relations,

organization and joints between these elements, allowing the company to be represented in a holistic and integrated perspective, in order to achieve the business objectives and facilitate decision-making processes"

The above definition brings together the most important elements provided by various authors: [13; 14; 15; 16; 17; 18; 19; 20].

2.3.1 Enterprise Architectures and Elements in Common

In recent years, several researchers have proposed enterprise architectures, among which stand out: CIMOSA [6; 21], GIM-GRAI [22], PERA [12; 23; 24], GERAM [12; 25], IE-GIP [7; 26; 27], TOGAF-ADM [18; 28], ARDIN [2; 10; 29] and ARIS [30].

The common elements that handle these enterprise architectures are: methodology, framework and language modeling. The following describes the use of each of these elements: the definition of a methodology facilitates the implementation of the architecture [15]; the framework allows a graphic and simple structure of the elements that make up the enterprise [16] and how these elements are related [15], according to the standard ISO/CEN 19439 [31], the framework should be composed of views and life cycle phases, and most enterprise architectures follow this approach; furthermore modeling language allows to model, organize and understand the relationship between elements of the enterprise. The enterprise modeling usually is developed in a modular way but at same time these modules must be integrated to ensure a holistic view of the company [6].

3 Conceptualization of Enterprise Collaboration

Nowadays, companies do not compete individually. Supply chains or supply networks compete with each other in the quest to increase profits and create greater added value. Therefore, the degree of integration between the partners that make up these supply chains (customers and suppliers) is growing. This degree of integration can be achieved through collaborative arrangements that ensure the alignment of individual plans in the pursuit of a goal or overall plan. It is here, where the enterprise collaboration becomes as a tool that allows partners of the supply chain a collaborative joint decision-making based, allowing coordinate and synchronize the global activities with the aim of satisfying the customer and increase the profits.

3.1 Definition of Enterprise Collaboration

From the definitions of [32; 33; 34; 35; 36; 37; 38; 39], we define the enterprise collaboration as:

"*A joint process between members of the supply chain, where the decisions are made jointly, based on the information shared and exchanged on a multilateral form, achieving coordinate and synchronize joint activities to meet customer requirements and achieve efficiency joint processes to generate mutual profits".*

3.2 Description Collaboration Process

Once the trading partners agree to participate in a collaborative process, this process will start. According to [40], collaboration process consists of six activities: 1) Definition, 2) Local domain planning, 3) Plan exchange, 4) Negotiation and exception handling, 5) Execution and 6) Performance measurement. However, in this generic process is not taken into account a crucial aspect in order to determinate an efficient collaboration, this aspect refers to how to share benefits equitably to ensure the stability of the collaboration [41]. The solution to this is provided by [39], who proposes the definition of a system of compensatory payments, which could be agreed on the definition phase of the negotiation and can be implemented when the results will be evaluated. Another aspect not covered by the process defined by [40], is the need for feedback between the partners once the process of collaboration in the specified horizon has been completed, moreover, the plan must be reviewed and modified if it is necessary.

4 Relationships between the Fields of Enterprise Architecture and Enterprise Collaboration

There are several studies that propose different types of enterprise architecture; however, most are designed to be implemented in an individual company. These architectures do not take into account the new collaborative environments in which companies are obliged to take part to ensure their survival. The architectures that are structured to be implemented in collaborative environments are: ARDIN EVEI (ARDIN Extension for Virtual Enterprise Integration) [2], CFCEBPM (Collaboration Framework for Cross-enterprise Business Process Management) [1], E2AF (Extended Enterprise Architecture Framework) [4] and VECCF (Virtual Enterprise Chain Collaboration Framework) [3]. As mentioned above, the main architectures have in common three elements: framework, methodology and modeling language, therefore we analyzed these three elements to the four above-named collaboration architectures.

4.1 Analysis of Enterprise Architecture Frameworks in the Context of Collaboration

The four architectures analyzed provide a framework; the Table 1 makes a comparative analysis of relevant elements and data of each of these proposals. The four frameworks work with views and some of these views are common among the frameworks; however, the levels posed by each framework are very different from each other.

The levels that define the framework proposed by [2], correspond to the life cycle phases defined by GERAM [25] and the ISO/CEN 19439 [5], in absence of the identification phase. The levels that define the framework proposed by [1], can be

Table 1. Comparative analysis of proposals in the context of collaboration frameworks

ELEMENTS AND DATA RELEVANT FOR THE COLLABORATION FRAMEWORKS		Framework name	Collaboration Framework for Cross-enterprise Business Process Management (CFCEBPM)	ARDIN extension for virtual enterprise integration (ARDIN-EVEI)	Virtual Enterprise Chain Collaboration Framework (VECCF)	Extended Enterprise Architecture Framework (E2AF)
		Year	2005	2003	2008	2006
		Authors	Adam, O.; Hofer, A.; Zang, S. Hammer, C.; Jerrentrup, M. and Leinenbach, S. (1)	Chalmeta, R y Grangel, R (2)	Choi, Y.; Kang, D.; Chae, H. and Kim, K. (3)	Schekkerman, J (4)
		It is based on the principles of	ARIS (Architecture of integrated Information Systems)	ARDIN[1] (Arquitectura de referencia para el desarrollo integrado)	MDA (Model Driven Architecture)	Framework de Zachman
Elements	Views	Global or collaboration	X			
		Function	X	X		
		Processes	X		X	X
		Information	X	X		X
		Resources		X		
		Decision		X		
		Service			X	
		Tecnology			X	X
		Information Systems				X
	Levels	Enterprise definition		X		
		Requirements definition		X		
		Design specification		X		
		Application description		X		
		Operation		X		
		Disolution		X		
		Collaborative business strategy	X			
		Engineering collaborative business processes	X			
		Implementation of collaborative business	X			
		Metamodel			X	
		Reference Model			X	
		Particular model			X	
		Contextual (Why?)				X
		Environmental (Who?)				X
		Conceptual (What?)				X
		Logical (How?)				X
		Physical (Whith What?)				X
		Transformational (When?)				X

[1] Spanish acronym of Reference Architecture for Integrated Development.

equated with the life cycle phases proposed by ISO/CEN 19439 [5], but in a general form, where the identification and definition of the concept can be matched with the collaborative business strategy; the requirements definition, design specification and description of the application correspond with the engineering collaborative business processes, and the operation corresponds with the implementation of a collaborative business. The levels defined in the framework proposed by [3], may also relate to the life cycle phases, but also are closely related with aspects of enterprise modeling, from a general level to a particular level. The levels that define the framework proposed by [4], match with most abstractions defined by [42], but this framework of collaboration includes the abstraction of "With What?", which seeks to represent solutions for each of the other views defined.

4.2 Analysis of the Methodologies in Enterprise Architectures in the Context of Collaboration

Only the proposals of [1] and [2] provide a methodology for the implementation of the collaboration architecture. The methodology of [1] includes five phases: Phase 1) Strategy partner analysis, Phase 2) Local TO BE concept, Phase 3) Global TO BE concept, Phase 4) Local implementation and Phase 5) Collaboration execution. Phases 1, 3 and 5 deal with joint collaborative processes, and phases 2 and 4 deal with local domain of each partner in the collaboration.

We miss two phases in this methodology: a preliminary stage of understanding and closeness between partners, in order to identify the strengths of collaboration and envisioning a future state of collaboration and a final phase to assess the results in collaboration and make the necessary reconfiguration if there were differences between partners about the results. According to the above, we propose the following phases: Phase 0) Knowledge and decision to collaborate and Phase 6) Feedback of results and reconfiguration or dissolution.

Regarding to the methodology proposed by [2], its structure is very similar to the previous one, but it also takes into account AS-IS processes, in this way, the current situation is understood by all participants and it is employed to search TO BE processes, in order to avoid the process definition based only on ideals.

These two methodologies (in content and form) are quite similar to the collaborative planning process discussed in Section 3.2, all the activities in this process, are included in the phases of the methodology, so conceptually, the methodology of enterprise architectures in the context of collaboration allows that the collaborative planning process can be conducted through a series of consecutive and related phases.

These methodologies can also be compared with the life cycle phases proposed in the individual enterprise architectures [21; 23; 25; 26], but have been adapted for collaborative environments, defining that activities should be carried out jointly and collaboratively and that activities should be conducted at the individual level in order to adapt processes and individual resources to the definitions and requirements established in the global domain.

4.3 Analysis of Modeling Languages in Enterprise Architectures in the Context of Collaboration

Researches by [1], [2] and [3], provide modeling language quite different from each other. By [1], is necessary to use tools that allow the visualization of the collaborative process and ensure a common understanding of collaborative processes between all companies and individuals involved in the process of joint business, so they propose to use a specific software (INTERACTIVE Process Modeler^{VR}), this tool provides an intuitive platform for Internet-based communication to register the business processes in an interactive and decentralized form. The employees are functionally responsible in the description of the business processes and establish an agreement between them through a virtual environment. The modeling language used in this software is BPML (Business Process Modeling Language), which they consider an appropriate exchange language. According to [3], they propose to use a platform-independent modeling architecture based on models (Model Driven Architecture - MDA), an initiative from OMG (Object Management Group), this platform is based on UML (Unified Modeling Language). Finally [2], propose the use of IDEF0 and GRAI nets to represent a general level of the different activities and decisions within the different business, they also propose using UML to describe the business process of the virtual enterprise and related information systems with a more specific detail level.

The existence of such different modeling languages makes it much more complicated to implement the interoperability of the various enterprise networks. However, interoperability between the companies that establish collaborative agreements is possible through the use of enterprise architectures that allow standardization and synchronization of joint processes and the integration of the key elements of global business.

5 Conclusions

At the present time, it is evident the interest that is taking the enterprise engineering through the use of enterprise architectures, both in the academic field as well as enterprise field. In recent years, interest is taking the enterprise engineering through the use of enterprise architectures is demonstrated by the large number of publications found in this respect, however, these methodological proposals are mostly directed and oriented to individual companies, which is not taking into account the growing and rapidly changing environments to inter-enterprise collaboration. The operation and organization of these collaborative networks should be structured and modeled through collaboration architectures designed with the purpose of supporting collaborative processes and their integration with information systems of partners involved in the collaboration. In this sense, some frameworks have emerged that seek to propose a general framework of collaboration, but these proposals fall short in the field of enterprise architecture, these do not meet all the elements identified as necessary for its correct implementation (framework, methodology and language modeling).

The analysis of elements of enterprise architecture in the context of collaboration had allowed us to identify: 1) All research proposes a framework, under the approach of views and levels. Although some views are common, the levels of each framework are very different, indicating different degrees of abstraction. 2) Only two research make a proposed methodology, these are closely related both to the collaborative planning process and the life cycle phases of the proposed enterprise architectures of individual enterprise, adapted to collaborative context. 3) Three investigations make different proposals of language modeling. The existence of such different modeling languages makes it difficult interoperability of various enterprise networks, but interoperability in the enterprise that establish collaborative agreements are facilitated by the use of enterprise architectures in the collaboration context.

We miss researches combining the disciplines of enterprise architecture and enterprise collaboration in a balanced way. Therefore, one possible research line that takes strength, following the analysis of the state of the art is to propose a collaboration architecture that meets all the necessary elements defined as in the field of enterprise architecture and also consider the methodological component of the business collaboration process, in order to facilitate the definition and conceptualization of collaboration architecture through collaboration architecture framework, the implementation of the collaborative process through an appropriate methodology understood by all partners in the collaboration and modeling language that facilitate understanding of the architecture of collaboration and allow interoperability of the collaborative processes.

References

1. Adam, O., Hofer, A., Zang, S., Hammer, C., Jerrentrup, M., Leinenbach, S.: A Collaboration Framework for Cross-enterprise Business Process Management. In: First International Conference on Interoperability of Enterprise Software and Application, Geneva (2005)
2. Chalmeta, R., Grangel, R.: ARDIN extension for virtual enterprise integration. The Journal of Systems and Software 67, 141–152 (2003)
3. Choi, Y., Kang, D., Chae, H., Kim, K.: An enterprise architecture framework for collaboration of virtual enterprise chains. The International Journal of Advanced Manufacturing Technology 35, 1065–1078 (2008)
4. Schekkerman, J.: Extended Enterprise Architecture Framework Essentials Guide. Institute For Enterprise Architecture Developments, IFEAD (2006), http://www.enterprise-architecture.info/index.htm
5. ISO 15704. Industrial automation systems - Requirements for enterprise-reference architectures and methodologies.: International Organization for Standardization (2000)
6. Kosanke, K., Vernadat, F., Zelm, M.: CIMOSA: Enterprise engineering and integration. Computers in Industry 40, 83–97 (1999)
7. Cuenca, L.: Marco arquitectónico para la propuesta IE-GIP. Extensión de la arquitectura CIMOSA. Aplicación a una empresa del sector cerámico. PhD thesis. Universidad Politécnica de Valencia
8. Molina, A., Panetto, H., Chen, D., Whitman, L.: Enterprise Integration and Networking: challenges and trends. Studies in Informatics and Control 16(4), 353–368 (2007)
9. Ortiz, A., Lario, F., Ros, L.: Enterprise Integration—Business Processes Integrated Management: a proposal for a methodology to develop Enterprise Integration Programs. Computers in Industry 40, 155–171 (1999)

10. Chalmeta, R., Campos, C., Grangel, R.: References architectures for enterprise integration. The Journal of Systems and Software 57, 175–191 (2001)
11. Vernadat, F.: Enterprise modeling and integration (EMI): Current status and research perspectives. Annual Reviews in Control 26, 15–25 (2002)
12. Williams, T., Li, H.: PERA and GERAM enterprise reference architectures in enterprise integration. Information Infrastructure Systems for Manufacturing, 1–27 (1998)
13. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Heidelberg (2009)
14. Arango, M., Londoño, J., Zapata, J.: Arquitectura empresarial- Una visión general. Revista Ingenierías Universidad de Medellín 9(16), 101–111 (2010)
15. Bernard, S.: An introduction to enterprise architecture. AuthorHouse, Bloomington (2005)
16. Cuenca, L., Ortiz, A., Boza, A.: Arquitectura de Empresa. Visión General. In: IX Congreso de Ingeniería de Organización, Gijón (2005)
17. Maya, E.: Arquitectura empresarial: un nuevo reto para las empresas de hoy. Centro de Investigación de las Telecomunicaciones. INTERACTIC: Interacción con la información (2010), `http://www.interactic.org.co/`
18. THE OPEN GROUP.: ARCHIMATE, The Power of Enterprise Architecture (2009), `http://www.archimate.org/en/home/`
19. Stelzer, D.: Enterprise Architecture Principles: Literature Review and Research Directions. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 12–21. Springer, Heidelberg (2010)
20. Schekkerman, J.: Enterprise architecture validation. Achieving business-aligned and validated enterprise architectures. Institute For Enterprise Architecture Developments, IFEAD (2004), `http://www.enterprise-architecture.info/index.htm`
21. Kosanke, K.: CIMOSA Primer on key concepts, purpose and business value (1996), `http://cimosa.cnt.pl/`
22. Chen, D., Vallespir, B., Doumeingts, G.: GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology. Computers in Industry 33, 387–394 (1997)
23. Rathwell, G.: PERA Enterprise Integration Web Site (2005), `http://www.pera.net/`
24. Williams, T., Rathwell, G., Li, H.: A handbook on master planning and implementation for enterprise integration programs. PERA Enterprise Integration Web Site (2001), `http://www.pera.net/`
25. IFIP.: GERAM: Generalised Enterprise Reference Architecture and Methodology. International Federation for Information Processing (1999), `http://dl.ifip.org/index.php/index/index`
26. Ortiz, A.: Propuesta para el Desarrollo de Programas de Integración Empresarial en Empresas Industriales. Aplicación a una Empresa del Sector Cerámico. Universidad Politécnica de Valencia (1998)
27. Cuenca, L., Boza, A., Ortiz, A.: Architecting business and IS/IT strategic alignment for extend enterprises. Studies in Informatics and Control 20(1), 7–18 (2011)
28. The Open Group (2011), `https://www.opengroup.org/index.htm`
29. Grangel, R.: Propuesta para el Modelado del Conocimiento Empresarial. PhD thesis Universidad Jaume I de Castello (2007)
30. Scheer, A., Schneider, K.: ARIS – Architecture of Integrated Information. Handbook on Architectures of Information Systems. International Handbooks on Information Systems 3, 605–623 (2006)
31. ISO/CEN 19439. Enterprise integration - Framework for enterprise modelling.: International Organization for Standardization (2006)

32. Stadtler, H., Kilger, C.: Supply Chain Management and advance planning. Concepts, Models, Sofware and Cases Studies. Springer, Heidelberg (2002)
33. Alarcón, F., Ortiz, A., Alemany, M., Lario, F.: Planificación Colaborativa en un contexto de varias Cadenas de Suministro: ventajas y desventajas. In: VIII Congreso de Ingeniería de Organización, Leganés, pp. 857–866 (2004)
34. Alarcón, F.: Desarrollo de una Arquitectura para la definición del proceso de Comprometer Pedidos en contextos de Redes de Suministro Colaborativas. Aplicación a una Red compuesta por Cadenas de Suministro en los Sectores Cerámico y del Mueble. PhD thesis Universidad Politécnica de Valencia (2005)
35. Petersen, K., Ragatz, G., Monczka, R.: An Examination of Collaborative Planning Effectiveness and Supply Chain Performance. The Journal of Supply Chain Management 41(2), 14–25 (2005)
36. Ribas, I., Companys, R.: Estado del arte de la planificación colaborativa en la cadena de suministro: Contexto determinista e incierto. Intangible Capital, 91–121 (2007)
37. Ribas, I., Lario, F., Companys, R.: Modelos para la planificación colaborativa en la cadena de suministro: contexto determinista e incierto. In: Congreso de ingeniería de organización, Valencia, pp. 1–10 (2006)
38. Dudek, G.: Collaborative Planning in Supply Chains. Supply Chain Management and Collaborative Planning. Springer, Heidelberg (2009)
39. Stadtler, H.: A framework for collaborative planning and state-of-the-art. OR Spectrum 31, 5–30 (2009)
40. Kilger, C., Reuter, B., Stadtler, H.: Collaborative Planning. In: Stadtler, H., Kilger, C. (eds.) Supply Chain Management and Advanced Planning-—Concepts, Models Software and Case Studies, pp. 263–284. Springer, Heidelberg (2008)
41. Audy, J., Lehoux, N., D'Amours, S.: A framework for an efficient implementation of logistics collaborations. International Transactions in Operational Research, 1–25 (2010)
42. Zachman, J.: A Framework for Information Systems Architecture. IBM Systems Journal 26(3), 454–470 (1987)

Effective Disaster Management: An Interoperability Perspective

Ovidiu Noran and Peter Bernus

Griffith University Australia, School of ICT
{O.Noran,P.Bernus}@griffith.edu.au

Abstract. Natural and man-made catastrophic events appear be steadily increasing in intensity and frequency. Proper preparation, response and recovery are essential if mankind and its vital systems are to cope with and survive large-scale disasters. Unfortunately, on account of historical and geographical reasons, the organisations responsible for delivering emergency response services often under-perform (or even fail), typically due to a lack of proper interoperation and collaboration. This paper attempts to analyse interoperability issues specific to disaster management and propose a way forward based on the advances in interoperability research, using an enterprise integration and architecture perspective in order to ensure a holistic approach.

Keywords: Interoperability, Emergency Response, Enterprise Integration.

1 Introduction

Mankind appears to rely on increasingly complex systems in order to survive and evolve. At the same time, the rate and force of natural and/or man-made disasters (whether triggered or augmented by climate change) appears to be on the rise. In this context, it is now essential to be able to promptly and effectively prevent, prepare for, respond to and recover from catastrophic events. Governments typically respond by putting in place suitable policies and organisations. However, disaster management organisations (DMOs) exist in a multi-dimensional continuum (e.g. time, life history, geographical location, governance level), that inherently results in independent evolution and thus heterogeneity. While organisational diversity is rather beneficial, it requires additional effort towards proper and effective collaboration [1]. Coping with large scale catastrophic events typically requires resources and capabilities beyond those of any individual organisation, making effective cooperation and interoperation essential. As has been seen on many unfortunate occasions, failure to achieve this in the case of disaster management organisations is likely to have dire consequences including preventable loss of property and human life.

This paper aims to analyse disaster management–specific interoperability problems and to propose ways to address them using the knowledge accumulated in state-of-the-art interoperability research. The analysis and proposals will be performed from an enterprise integration and architecture point of view promoting a holistic approach through a combination of frameworks aiming to cover all aspects deemed as relevant.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 112–121, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Current Issues in Disaster Management

The operation of Emergency Services is typically legislated at state, national and international levels (see [2-5]). Unfortunately however, merely instructing organisations to cooperate using high-level generic directives does not bring true collaboration and / or interoperability. The consequences are extended response times, confusion on the situation on the ground, dispute / competition as to who, where and when is in charge, difficulty in coordination with other teams' systems due to incompatibilities in infrastructure and difficulty in filtering / validating the flood of information generated during disaster events. For example, lack of consistency in the alert notice type and delivery format may delay warnings or flood the population with ambiguous / irrelevant messages [6, 7]. This leads to sub-optimal preventative action and response by the intended recipients and potential property and life loss.

Two main approaches are used to address the above-mentioned problems; they involve either centralisation (hierarchical command) or federalisation of emergency services. Irrespective of the approach used however, proper emergency response and cooperation has still not been achieved, as reflected in criticism by authors and community [8-10]. The use of military operations-style network-enabled capabilities as the backbone of disaster management [11] is valid only as part of the overall disaster management effort (e.g. during some life cycle phases of the disaster events).

It appears that in fact the root causes are the inadequate information and knowledge flow and quality between the participants [12, 13], the lack of trust, organisational confusion and competition fallacies. True and efficient collaboration is not possible unless the organisational cultures, processes and resources of the participants possess the required interoperability preparedness [14].

Another aspect that appears to have been addressed to a lesser extent is the life cycle of the DMOs, task force, other controlling entities (e.g. laws, standards etc) and the disaster event(s). The interoperability requirements will vary in this context.

Analogies with other domains may help provide potential solutions. For example, the DMOs' situation resembles that of commercial enterprises that need to cope with a global business environment requiring them to tackle projects beyond their own resources and knowledge. A typical response to this problem in the commercial domain is to set up or join Collaborative Networks (CNs) that act as 'breeding environments' [15] for Virtual Organisations (VOs – groups of companies that act as one for a limited time) and can be promptly created in order to bid for, win and complete projects requiring combined resources and know-how. Another analogy is that of allied armed forces that need to cooperate in crisis situations. Standardised agreements on language and systems and joint exercises are used in this case [16].

DMOs are situated somewhat in between these two cases as they may have full time / militarised and voluntary / reserve staff components, depending on the geographical location and local legal and administrative situation. Thus, successful concepts from the commercial and military areas can be reused provided due customisation is performed so as to fit the specific DMO setting.

The ‘communities of engineering practice’ concept described in [17] can also be applied to disaster management so as to facilitate experience sharing, collaborative learning, professional development and to kick start transnational and trans-cultural partnerships.

3 Interoperability Frameworks

Interoperability is understood here as enabling the exchange of information and its use but also as ability to perform a function on behalf of another entity [18, 19]. Previous research and practice have identified various aspects of interoperability. For example, DoDAF (Department of Defence Architecture Framework) [20], via its Levels of Information System Interoperability (LISI), defines manual, peer-to-peer, distributed, integrated and universal levels (in fact expressing a kind of interoperability maturity). The European Interoperability Framework (EIF) [21] defines organisational, semantic and technical interoperability. The Athena Interoperability Framework (AIF) [22, 23] defines data, processes, services and business. IDEAS (Interoperability Development for Enterprise Applications and Software) [24] defines three levels: data, application and business process. Whitman and Panetto [1] identify technical, syntactic, semantic and pragmatic interoperability levels. It appears that various frameworks assign different meanings to the same concept (e.g. level) and different names to the same concept (e.g. aspects / viewpoints). Some concepts appear to be orthogonal, as illustrated in Chen’s [19] framework based on EIF, IDEAS and ATHENA, which is also adding the concept of barriers. Some frameworks combine concepts (and approaches) in matrices or cubes. The following sections will attempt to employ the concepts / frameworks that appear to be most relevant / applicable to disaster management.

4 Disaster Management Interoperability

Reasoning about improving DMO interoperability raises several questions. To what extent is interoperability required? What components / aspects of the participants need to interoperate? How can it be ensured that all necessary aspects are covered and interoperability is preserved over time as all participants evolve?

As each disaster event is unique, there is no ‘optimal’ DMO interoperability level to fit all crisis situations. At a minimum, the participating organisations should display compatibility, so at least they don’t hinder each other’s operations (see Fig.1). Full integration may not be desirable in this case as it could imply that the DMOs cannot function independently at full capacity. In an emergency situation, it is quite possible that one or several participants (or components thereof) could be affected and may even cease to function; the rest must be able to continue without significant performance loss (e.g. similar to the ARPANET resilient network concept [25]). Resilience and agility are essential in this case. Even if a central point of command (Emergency Command Centre) was secure and unaffected by the disaster event(s), the coordination provided by it could be severely affected by communication breakdown. The organisations involved should be able to continue within acceptable operating parameters in such scenarios.

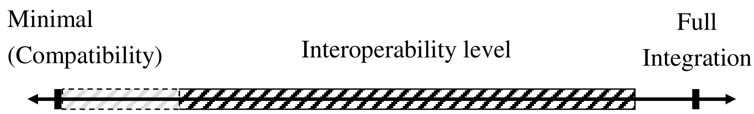


Fig. 1. Desired interoperability level range (based on [1])

The suitability of an interoperability improvement approach must be analysed in view of the framework components presented in the previous section. ISO14258 [26] establishes several ways to achieve interoperability: integrated (common format for all models), unified (common format at meta level) and federated (no common format, with participants negotiating an ontology on the fly so as the meaning of models is understood the same way). In the case of DMOs, it appears that the unified approach is most suitable as both full integration and federalisation did not seem to succeed and achieve the desired results due to organisational heterogeneity of DMOs and the impossibility to properly negotiate in the limited time available in the case of a disaster event. The unified approach assumes that an ontology is negotiated in advance. This requires that the DMOs spend time together prior to disaster events in order to agree on meanings associated with the concepts used to exchange knowledge. Once this is achieved, the semantic requirements for proper interoperability should be promptly met in the task forces formed by the participant DMOs.

The technical aspect of interoperability (and associated barrier [19]) appears to be considered the most straightforward to solve. However, disaster management task forces may comprise interstate/international organisations relying on different infrastructure and thus technical details may become the stumbling block if not given sufficient prior attention. This may relate to both software and hardware aspects.

The pragmatic aspect of interoperability as defined in [1] relates to the willingness, commitment, and capacity of the participants to interoperate. Although this aspect is mandated by governments, the human side needs attention prior to task force formation so as to allow gaining trust and knowledge of the other DMOs.

The syntax interoperability aspect applied to DMOs importantly includes 'human data exchange formats' in addition to machine ones - for example, the adoption of English as the language spoken by an emergency task force comprised of international DMOs. This appears to be reasonably easy to achieve; however, syntactic interoperability does not solve the semantic aspect, i.e. correctly understanding the intended meaning of the information exchanged. In the case of humans, non English-speaking background crews may understand or express concepts in an unintended way – which in the case of DMOs may have tragic consequences.

'Cultural interoperability' [1] - whether organisational, local or national - is one of the hardest obstacle to overcome. The only current true solution in the authors' opinion is regular immersion of the participant organisations in each other's cultures. This will facilitate the transfer / conversion of tacit and explicit knowledge between the participants, which is one of the most important goals of interoperability. Although cultural interoperability is not limited to language, an analogy can be made to human language translators; thus, an accurate translation can only be performed by a person that has a thorough knowledge not only of the languages (and associated jargons) but also of the philosophy / cultures of the nations speaking those languages.

Good translators would have ideally lived for some time in all countries speaking the languages they attempt to translate. In the universe of discourse of a company, the 'translators' could be the enterprise architects who are required to speak the technical jargon but also understand the culture of all departments (e.g. business, management, engineering and information technology). A good (enterprise) architect would have multiple qualifications and spend time in each department, gaining a deep understanding of all company areas, being able to bridge them and thus assist the enterprise integration effort.

Organisational interoperability is an important aspect in disaster management as participants may often have different organisational and management structures. The issues identified by Chen [19], based on the EIF framework (namely responsibility, authority and type of organisation) can all impact heavily on the functionality of the disaster management task force. In a crisis situation, the roles (mapping of the human resources onto the decisional structure) and hierarchy must be clear to everyone from the start so that the task force can focus on managing the disaster event rather than spend critical time figuring out its own modus operandi (who does what, who is in charge etc). Modelling of the interoperability of emergency agencies involved in Tsunami warning in Australia [27] found that the organisational aspect (using synchronous processes rather than asynchronous ones) was an impediment that rendered the warning system dysfunctional. While pockets of the organisation were aware of the problem, the lack of joint grounded experience created a lack of mutual understanding in the process. This situation was predicted by Bernus [28] who used situation semantics to prove that ontological theories alone are insufficient for achieving interoperability.

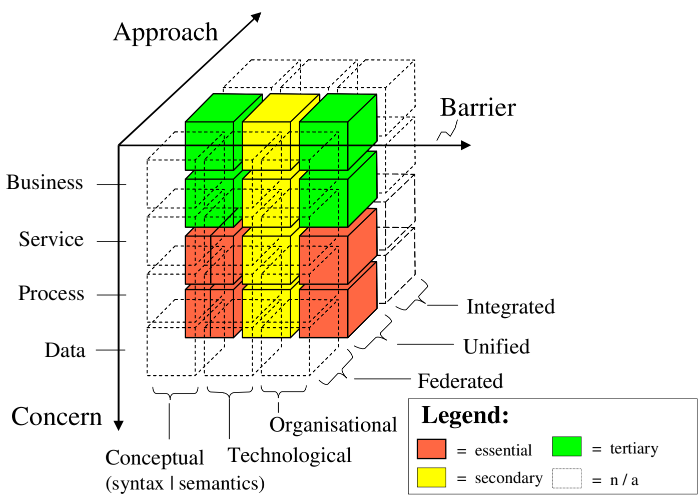


Fig. 2. Interoperability issues and approach in disaster management (based on [19])

Using the interoperability concerns identified by the ATHENA project, the data and process aspects appear to be the most relevant in disaster management. Thus, the ability of task force participants to extract and exchange data from potentially heterogeneous data repositories is paramount to being aware of the conditions on the ground and to avoid sending personnel into an unknown and potentially life-threatening situation. The high volume and erratic reliability of data present in a disaster event will compound this problem. Prior agreements on data format and especially on meaning are essential. Process interoperability in the author's opinion concerns both the capability to perform joint operations - which underpins the entire task force concept – but also to 'take over' and perform a process instead of a participant that may have been temporarily or permanently disabled. Again, prior agreements but also joint practice on the ground should be performed in order to address this important interoperability concern. The service aspect of interoperability concerns the possible interconnection of the services provided by the DMOs so as to form a more complex service and involves syntactic and semantic aspects that should be solved well in advance to the formation of a task force. The business interoperability aspect refers to organisational culture and management style of the interacting DMOs (addressed in the cultural interoperability section above).

Fig. 2 shows the proposed relevance of interoperability aspects in the case of disaster management using the framework presented in [19]. The framework has been selected because it combines concepts of several other interoperability frameworks. Note, however, that it is not 'complete' - while cultural interoperability may be partly represented in the organisational aspect, other important aspects, such as life cycle are not adequately represented. Typically, in order to ensure that all necessary interoperability aspects are covered, it is necessary to use a combination of frameworks. In addition, the modelling framework of a generalised (i.e. abstracted from several others) architecture framework can provide a more complete viewpoint repository that can be used to model and select relevant interoperability aspects for a specific project type, including disaster management.

5 Enterprise Architecture, Life Cycle and Disaster Management

The tasks involved in reaching and maintaining EI are on-going, complex, and often involve ill-defined constraints and uncertainty. Enterprise Architecture (EA) holds the promise to help stakeholders manage the EI effort in a consistent and coherent fashion, so as to achieve and maintain integration within and between organisations. For this reason it has been considered that the use of EA artefacts and methods can complement the improvement of DMO interoperability. To illustrate this point, we have selected ISO15704 Annex A (GERAM[1], see [29] for details), a generalised enterprise architecture framework (AF) based on several mainstream AFs. Among other advantages, the modelling framework (MF) of GERAM's reference architecture (GERA) allows representing all other aspects in the context of life cycle.

The left hand side of Fig. 3 shows how the typical disaster management activities [3] can be mapped to the life cycle phases of a disaster event (represented using the GERA life cycle dimension). Furthermore, in the figure it can be seen that the relevance and applicability of the aspects proposed by the reviewed interoperability frameworks are in fact depending on the specific life cycle phases of the disaster

[1] The Generalised Enterprise Reference Architecture and Methodology.

event and the DMOs' actions to address them. Therefore, the use of AF artefacts does promote and enrich the understanding of the interoperability problems specific to disaster management. Further mappings have been performed, involving other aspects of the GERA MF. They are not shown here due to space limitation constraints and will be disseminated in other papers.

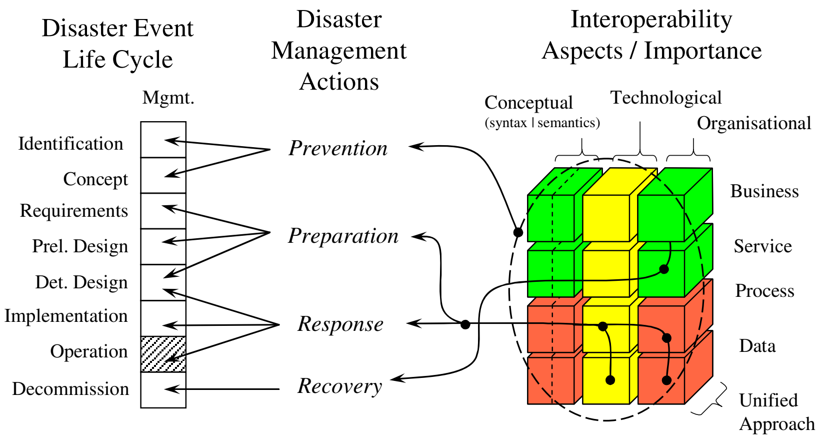
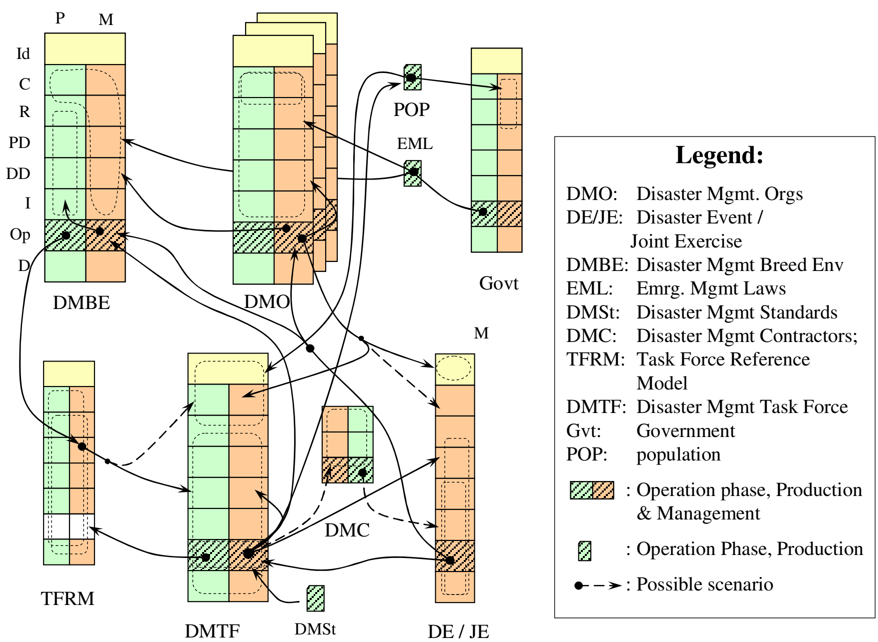


Fig. 3. Disaster event mapping and modelling using a GERA-based formalism (left) and an interoperability framework (right)

Reviewing the current interoperability and disaster management body of knowledge it has become clear that while at some levels interoperability can be promptly and significantly improved, the conceptual (especially semantic side) and human-related interoperability aspects such as trust, management, organisational and local culture must be addressed in advance. In addition, organisations evolve and thus the solutions found must be regularly reviewed. For this reason, the life cycle and management vs. mission accomplishment views of GERA have been employed in Fig. 4 in order to a) illustrate and propose a solution in the form of a 'breeding environment' approach and b) analyse the interoperability requirements in this context.

Thus, Fig. 4 shows how DMOs can form a so-called 'disaster management breeding environment' (DMBE). This structure allows participants to work together (whether in same space and/or time or not) in order to iron out critical interoperability issues well in advance, rather than wasting precious time during disaster events. For example, agreements can be reached and interfaces built if required to bridge technological and technical aspects such as infrastructure gaps. The conceptual syntactic interoperability can be addressed by agreeing on and upholding standardised formats. Semantic interoperability can effectively be addressed in this case by the personnel of participating DMOs spending time together physically (in the same location / time) and immersing in each other's cultures as previously described in this paper. This immersion is seen by the author as the only currently feasible way to also address the organisational, business and cultural interoperability, which together with the semantic aspect represent the ultimate goal of the entire interoperability effort.

The arrows between the entities shown in Fig. 4 represent amalgamated interoperability requirements, (detailed in further diagrams and omitted due to space limitations) in respect to importance, aspects, levels etc identified in interoperability frameworks. Thus, for example 'interoperation' of the population POP with the Government Govt is important as may result in changes to legislation and DMO and DMBE interoperability requirements. However, interoperability between DMOs within the disaster management task force (DMTF) and between the DMTF and POP is paramount as it will directly decide the amount of lost property and casualties. Past (and unfortunately often tragic) experience has shown that the two major goals of the disaster management interoperability improvement effort partly reflected in Fig. 4 should be a) whether POP receives, understands, believes and uses DMTF warnings and directives and b) that DMTF participants can properly interoperate during the disaster event DE.



Life cycle phases: Id: Identification; C=concept; R=requirements, PD=preliminary design, DD=detailed design, I=implementation, Op=operation, D=decommissioning. *Other aspects:* P=Production / Service, M=management

Fig. 4. Interoperability in the context of a Breeding Environment and Task Force solution

The validity and effectiveness of the DMBE concept can be tested by joint exercises (JEs) simulating DEs; this will also allow practical and perhaps still uncovered interoperability problems to surface and be tackled. Lessons learned from previous DEs / JEs should be structured in reference models (TFRMs) to preserve / reuse acquired knowledge and to decrease the response time. Importantly, the resilience of DMTFs created can be also assessed in JEs through various scenarios; corrective action can then be taken to improve preparedness towards real crisis situations.

6 Conclusions and Further Work

Disaster management interoperability related issues are best addressed in advance and in a holistic manner so that when a disaster event occurs an efficient, collaborative task force can be promptly put together. Using state-of-the-art research results in interoperability and enterprise architecture, the paper has analysed interoperability issues specific to disaster management and has reached several conclusions. Thus, the interoperability level of DMOs should best not reach full integration so as not to affect resilience. The cultural / organisational and semantic aspects of interoperability, as the most important and difficult to solve, should be tackled in advance, allowed time and be periodically reviewed. The life cycle of the participants, task force and disaster event(s) must form the background context to the entire interoperability improvement effort. It is also proposed that the adoption of breeding environments and joint exercises concepts customised for disaster management will greatly assist with all the interoperability aspects described in this paper.

Further research will concentrate on testing and validating the findings with DMO stakeholders within several case studies.

References

1. Whitman, L., Panetto, H.: The Missing Link: Culture and Language Barriers to Interoperability. Annual Reviews in Control 30(2), 233–241 (2006)
2. Government of South Australia. Emergency Management Act 2004, http://www.legislation.sa.gov.au/LZ/C/A/EMERGENCYMANAGEMENTACT2004.aspx
3. Australian Government. Attorney's General's Office - Emergency Management in Australia (2011), http://www.ema.gov.au/
4. Federal Emergency Management Agency. National Response Framework (2011), http://www.fema.gov/pdf/emergency/nrf/about_nrf.pdf
5. United Nations International Strategy for Disaster Reduction Secretariat (UNISDR). Hyogo Framework for Action 2005-2015: Building the resilience of nations and communities to disasters, http://www.preventionweb.net/files/1037_hyogoframeworkforactionenglish.pdf
6. Queensland Floods Commission of Enquiry. Submissions to the Enquiry (2011), http://www.floodcommission.qld.gov.au/submissions/
7. Victorian Bushfires Royal Commission. Submissions to the Enquiry (2009), http://www.royalcommission.vic.gov.au/Submissions/View-Submissions
8. Clark, J.L.: Practical aspects of federalizing disaster response. Critical Care 10, 107–113 (2006)
9. Wiese, C.R.: Organizing Homeland Security after Katrina: is adaptive management what's missing? Public Admin Review 66(3), 302–318 (2006)
10. 't Hart, P., et al.: The Politics of Crisis Management: Public Leadership under Pressure. Journal of Homeland Security and Emergency Management (2005)

11. von Lubitz, D.K.J.E., Beakley, J.E., Patricelli, F.: Disaster Management: The Structure, Function, and Significance of Network-Centric Operations. Journal of Homeland Security and Emergency Management 5(1), Art. 42 (2008)
12. Wickramasinghe, N., von Lubitz, D.K.J.E.: Knowledge Based Enterprises: Theories and Fundamentals. Journal of Homeland Security and Emergency Management 5 (2007)
13. Prizzia, R., Helfand, G.: Emergency preparedness and disaster management in Hawaii. Disaster Prevention and Management 10(3), 163–172 (2001)
14. Kapucu, N., Arslan, T., Demiroz, F.: Collaborative emergency management and national emergency management network. Disaster Prevention and Management 19(4), 452–468 (2010)
15. Camarinha-Matos, L., et al.: Collaborative networked organizations - Concepts and practice in manufacturing enterprises. Computers and Industrial Engineering 57(1), 46–60 (2009)
16. NATO. Interoperability for joint operations (2006), http://www.nato.int/docu/interoperability/interoperability.pdf
17. Molina, A., Aca, J., Wright, P.: Enterprise integration and networking: issues, trends and vision. In: Knowledge Sharing in the Integrated Enterprise – Interoperability Strategies for the Enterprise Architect. IFIP, Toronto Canada (2004)
18. Vernadat, F.: Enterprise Modelling and Integration: Principles and Applications. Chapman & Hall, London (1996)
19. Chen, D.: Framework for Enterprise Interoperability (2006), http://www.fines-cluster.eu/fines/jm/Download-document/53-Framework-for-Enterprise-Interoperability-Chen.html
20. DoD Architecture Framework Working Group. DoD Architecture Framework Ver 1.0 (2004)
21. EIF, European interoperability framework for pan-European eGovernment services. Luxembourg: Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC) (2004)
22. ATHENA State of the art of Enterprise Modelling Techniques and Technologies to Support Enterprise Interoperability. Deliv D.A1.1.1 2004 (2011), http://www.athena-ip.org
23. Berre, A.-J., et al.: The ATHENA Interoperability Framework (2006), http://www.txtgroup.com/newsletter/attachment/Athena_Paper.pdf
24. IDEAS. IDEAS Project Deliverables (WP1-WP7), Public reports (2003), http://www.ideas-roadmap.net
25. Heart, F., et al.: ARPANET Completion Report, Bolt, Beranek, Newman, Editors, Burlington, MA (1978)
26. ISO, ISO14258 Industrial Automation Systems – Concepts and Rules for Enterprise Models (2005)
27. Zhao, Z.: Process Modelling for Tsunami Warning System through Enterprise Engineering Methodology (2005), http://www.ict.griffith.edu.au/~bernus/Tsunami.pdf
28. Bernus, P.: Enterprise Models for Enterprise Architecture and ISO9000:2000. Annual Reviews in Control 27(2), 211–220 (2003)
29. ISO/IEC, Annex C: GERAM, in ISO/IS 15704/Amd1-2005: Industrial automation systems - Requirements for enterprise-reference architectures and methodologies (2005)

Design of Large-Scale Enterprise Interoperable Value Webs

Wout Hofman

TNO, P.O. Box 5050, 2600 GB, The Netherlands
Wout.hofman@tno.nl

Abstract. Still a lot of enterprises are faced with the issue of interoperability. Whereas large enterprises are able to implement the required technology, SMEs (Small and Medium sized Enterprises) face challenges as they lack knowledge and budget. Enterprises have defined their specific semantics and business processes that have to be integrated with those of others for acting in service networks or value webs. Solutions do not yet offer a plug-and-play interoperability environment, although these solutions support open standards. From a technical perspective, an enterprise can implement a Service Oriented Architecture (SOA), an Event-Driven Architecture (EDA), a combination of both called EDSOA, or messaging. The Semantic Web Architecture in which data is referred to is yet another architectural approach to interoperability. This paper defines an interoperability ontology decomposed in (1) business concepts like value propositions and value exchange represented by a business interoperability ontology and (2) their architectural grounding to technology and like XML Schema Defintions. IT tools are briefly discussed that support the specialization of the business interoperability ontology to for instance logistics and to manage the architectural grounding. The proposed interoperability ontology needs to support large-scale implementation of enterprise interoperability.

Keywords: Enterprise interoperability, large-scale infrastructure, value web, logistics.

1 Introduction

In a service economy [19], enterprise interoperability is of growing importance [3]. Enterprises need to exchange and share data as integral part of their value propositions [2]. In most solutions, business documents are exchanged either on paper or using electronic formats like EDI messages (EDI: Electronic Data Interchange) and XML Schema. Service Oriented Architecture (SOA, [1]) and Event-Driven Architecture (EDA, [4]) combined to an Event-Driven Service Oriented Architecture [4] are other interoperability solutions. All architectural approaches offer mechanisms and patterns, but are not able to specify semantics. In the meantime, semantic technology is introduced for creating what is called a Semantic Web [5], basically for improving the precision and recall of content queries: getting complete and correct results upon one's search query in a fast growing amount of data based on underlying

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 122–135, 2011.
© Springer-Verlag Berlin Heidelberg 2011

semantic concepts and supporting technology [6], [7], and [8]. Whereas the Semantic Web considers content retrieval as Linked Open Data (LOD, [13]), enterprise interoperability deals with data and process synchronization of different legal entities [9]. It is accepted that business and data modelling is of importance to interoperability [10].

However, large scale enterprise interoperability is not yet implemented whilst closed systems based on bilateral agreements of technical standards are constructed [14]. These closed systems construct a value web in which some organizations are able to share data electronically, whereas others still operate with paper. Many SMEs are not able to invest in these different customer and supplier solutions. Eventually, the fact that not all enterprises are interoperable imposes different procedures on those that are, namely an electronic and a paper based procedure. One of the reasons of having closed systems, is the lack of a clear meta-model specifying the grounding [15] of semantics to several architectural solutions supported by tools like an Enterprise Service Bus (ESB) for large scale interoperability against reasonable costs. In those cases, where semantics is specified separately from technical solutions [21], semantics has to be copied to become re-usable, thus introducing potential inconsistencies based on interpretation differences. Existing standards have been extended to increase their applicability with inclusion of semantics, e.g. OWL-S [11], which still does not solve the issue of consistency between different solutions based on those standards.

This paper presents a first version of such an interoperability ontology for value propositions and value exchange with an architectural grounding to various technological solutions [14]. We call this ontology 'interoperability ontology'. It is decomposed into a 'business interoperability ontology' and an ontology for managing architectural grounding to a particular technical solution. The business interoperability ontology can be specialized to application areas like logistics. Furthermore, we show how tools can support the specializing the interoperability ontology for an application area. Finally, some conclusions and issues for further research are identified.

This paper takes a service oriented approach. Value propositions are equal to business services, i.e. services provided by an enterprise at business level. However, these business services are not necessarily implemented by IT service, i.e. web services. Currently, business most often share information in an asynchronous manner based on messaging (see section 3.1).

2 Business Based on Value Propositions

This paper takes value propositions [2] as the core concept for business common to enterprises. Value propositions define the value that can be offered by an enterprise to another enterprise or person and lead to actual value exchange. Thus, a loosely coupled organizational network or value web can be constructed in which relations between enterprises and/or persons are established at runtime and are dynamic. This section models the core concept 'value proposition' and its related concepts by a business interoperability ontology. The business interoperability ontology is an upper ontology that can be specialized further to an application area like logistics as shown in this section. Such a specialization constructs one or more so-called lower ontologies. The next section presents the technical support of data sharing for this business interoperability ontology.

2.1 Modelling Business Concepts as Ontology

Information is shared amongst enterprises for different purposes. This paper takes the purpose of enterprises exchanging value at business level. This section defines 'business activity' and 'value proposition' as the shared concepts for value exchange between enterprises. The concept 'actor' is used in this section to represent 'enterprise', 'organizational unit', or 'government organization'.

The concept 'business activity' is based on concepts 'event' and 'action' from a business perspective. 'Event' can be defined as 'an operation on an object with given properties at a specific time or within a time interval' [16] and 'action' is 'the intentional behaviour of an actor to achieve a certain goal' [17]. These definitions imply that an event may lead to an action, i.e. an actor can trigger or execute an action based on an event initiated by another actor, As an action can be performed by an actor, the actual performance of that action brings value to a customer against known costs. A business activity equals an action at business level. The concept 'business activity' is generic to a particular application area like 'logistics', 'finance', and 'government' as it can be provided by many actors. It is defined as an action that can be provided by one or more actors. 'Capability' [15] and 'state transition' [24] are synonyms for business activity. To be able to exchange value, resource capacity needs to be allocated and utilized, e.g. a truck and its driver, and exchanged a part of value exchange. A resource is a person and/or real world object required by or exchanged between two actors.

A business activity has a provider and a customer. A customer can request the execution of a business activity by a provider by formulating a 'goal' [15]. Information needs to be shared amongst actors for resource allocation; for instance by exchanging business documents that we will call 'interactions'. A business transaction defines all information shared with interactions between a customer and provider for the execution of a business activity according to a particular choreography for those interactions [19]. The choreography can be decomposed in three transaction phases that govern actual value exchange. As in [12] and [23], the following transaction phases with a certain sequencing logic are distinguished: quotation that results in agreement on execution or is cancelled, execution in which value is actually exchanged, and cancellation. Cancellation can not succeed the quotation, but can also be performed during execution to handle with exceptions, e.g. a container can not be loaded onto a vessel if that container is too late at the premises of the proper stevedore. The concept 'interaction type' is an event with a function in a transaction phase. It is synonym of the concept event for information sharing in REA [18]. An interaction is a function within the governance of value exchange and used to share state changes in information reflecting the value exchange.

Figure 1 shows the above mentioned concepts as ontology. It presents the concepts and their associations that together form a business interoperability ontology. Figure 1 also shows that a business activity can take input from and produce output in a place. A resource specifies the input or output. It has to be available in that place for input by a business activity at a certain time and can be available in an output place at another time, possibly in a different form if the business activity is 'production'. This paper defines 'value proposition' [2] as an instance of a business activity. By stating that an actor supports business transactions of a business activity, implies that that actor is the provider of value proposition. Whereas this paper defines two roles namely customer and provider, the role of an actor most commonly refers to the name of a (set of) business activity(ies), e.g. a carrier offers value propositions of business activity 'transport'. This paper only distinguishes the roles of customer and provider and value propositions as the basis for value exchange.

The actual value exchange is based on information sharing and interactions by 'business transaction'. The state of value exchange is specified by the interactions of a type with a function in a transaction phase. By modelling the sequence in which interactions are exchanged (modelled by 'previous'), one is able to reconstruct the state changes governing value exchange. Business transactions and interactions are more or less considered to be operational, i.e. they are more dynamically changing with higher frequency than instances of the other concepts.

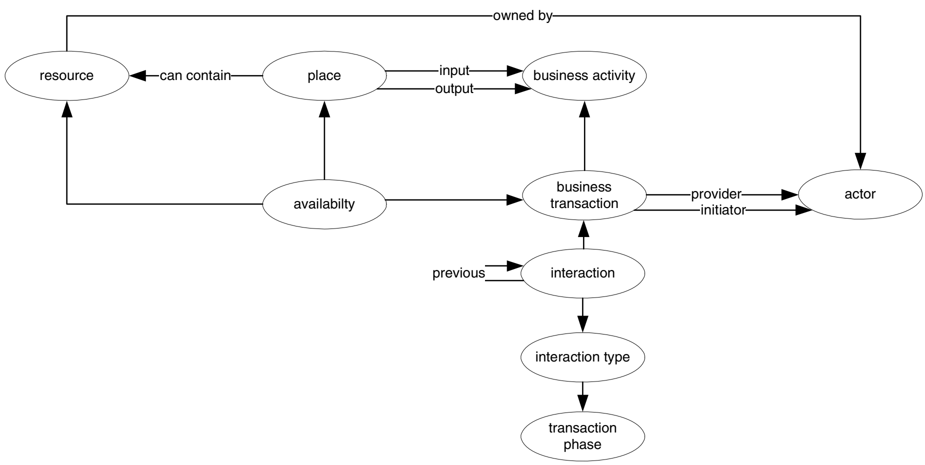


Fig. 1. Business interoperability ontology

The business interoperability ontology can be further specialized to an application area as shown hereafter. The ontology can also be applied to virtual resources like multimedia. In those cases, 'place' represents a state in which these resources are. There are also application areas in which the number of states can be large, e.g. government applications where the value of each concept can represent a separate state. In those application areas, 'state' or 'place' can be modelled as part of the precondition of a business activity. A precondition validates whether or not a concept specifying a resource is 'available' for that particular business activity, e.g. a person has a driver's licence [20].

The business interoperability ontology, which is an upper ontology, can be specialized in various ways to a lower ontology. One can construct an ontology for a specialization of 'business activity', but also for specialization of 'resource'. By allowing different types of specializations, e.g. to 'business activity' and 'resource', one is able to construct combined (lower) ontologies in which for instance different specializations of ‘business activity’ refer to identical specialization of ‘resource’. One can for example construct a set of ontologies for container transport by only importing ‘containers’ as ‘resource’ for specializations of ‘business activity’. Another approach would be to create one lower ontology for all types of logistic services and resources and adding rules for specific types of resources. Whereas in the first approach, networked ontologies are created that need to be managed, in the second approach complexity is in managing all rules. At this moment, networked ontologies seems to be a better approach, if tool support is sufficient (see further).

2.2 An Example: Logistics

Logistics consists of several actors that utilize each others resources to transport goods from one physical location to another, each with its particular value proposition or logistic service. Logistics can be quite complex, involving a variety of actors. International container transport by sea for instance consists of transport by truck to a port (pre-carriage), loading a container in a vessel, sea transport, discharge of the container in a port of destination, short sea shipment from that port to another port, and transport by truck to a final destination, all arranged and executed by different enterprises. Shippers can also have regional warehouse close to a port, from which goods are regionally distributed by trucks. This paper only presents ‘transport’ as a specialization of ‘business activity’ for specifying logistic services, whereas ‘transshipment’ and ‘storage’ are examples of relevant logistic business activities. First of all, we define ‘transport’ as activity and secondly we present an ontology for this particular activity.

2.2.1 The Activity ‘Transport’ for Logistic Services

‘Transport’ is a specialization of ‘business activity’ and is characterized by a number of parameters like:

- The cargo type to be transported, e.g. containers, pallets, boxes.
- The minimal/maximal number of units of that cargo, e.g. maximal 10 pallets.
- Physical characteristics of the cargo including minimal/maximal measurements, dimensions, and temperature conditions during transport. Container measurements are represented by a container type with sometimes over dimensions, e.g. for transporting yachts.
- Cargo descriptions describing in general terms the nature of the cargo, e.g. electronics or textile.
- An indication whether or not the cargo is dangerous with characteristics, which could require extra permits.
- From-/to-locations.
- The duration that can be expressed differently like a maximum, a guaranteed, an expected, etc.

According to the previous ontology (Figure 1), cargo is a specialization of ‘resource’ that can be in a particular place at a given time (‘availability’). A customer can initiate a business transaction by formulating that cargo is to be transported to another place and expected to be in that place at a certain time. Only those instances of business activity ‘transport’ can fulfill these conditions that have pre- and post-conditions matching a customer goal. These pre- and post-conditions actually form value propositions that can be expressed in natural language as:

- The pre-condition defines the resources that are input:
  - The cargo offered by a customer is part of the set of cargo types defined by a value proposition.
  - The physical dimensions of the cargo offered must be in a certain range defined by a value proposition, e.g. weights below 1 kg, between 1 and 10 kg, etc.
  - The number of units must be in a range.
- Post-condition: the cargo offered by a customer is transported by a service provider to a place of delivery according to an estimated duration and costs. In fact, the post-condition is the offer selected by a service provider to meet a customer goal.

Note that the instances of ‘place’ and their relation to ‘value proposition’ as instances of ‘business activity’ define geographical areas for those value propositions, e.g. the regions between which a value proposition is offered. It is also feasible that certain logistic services are based on a time schedule, meaning that logistic service with an instance of ‘transport means’ is related to instances of ‘place’ with ‘availability’. Connecting places to ‘transport’ activities defines a so-called hub-spoke transport networks, e.g. a logistic service is in a region covering various countries or part of countries between the Netherlands and northern part of Italy via Austria and Germany or between the EU and the Asian Pacific.

2.2.2 Modelling ‘Resources’ for Transport

A transport interoperability ontology is the specialization of the business interoperability ontology introduced in section 2.1 and is the basis for sharing data for transporting cargo from one location to another. The OWL document of Figure 1 is further specialized for business activity and resource (Figure 2). ‘Resource’ is specialized to ‘transport means’ and ‘cargo’. Transport means are further specialized to vessel, truck, barge, airplane, and train. Each of these transport means has a specific transport modality (sea, rail, inland waterways, road, air) and its own characteristics, e.g. a truck has a license plate and a vessel a Radio Call Sign. Cargo can be further specialized in equipment, general cargo, and bulk cargo. Equipment, e.g. containers, is re-used for transporting purposes and requires additional management functions called for instance ‘container repositioning’. General cargo encompasses cargo with packaging that is in principle not managed per individual packing unit for re-use (e.g. cartons, boxes, and pallets). Additionally, a number of properties and constraints need to be formulated, e.g. a transport means has a modality (a specialized transport means like a truck should have modality ‘road’) and a property stating that a certain transport means can only be used for container transport. A particular vessel or truck is an instance of ‘vessel’ and ‘truck’.

Figure 2 shows an initial draft of one transport interoperability ontology. The paper already discussed options of constructing a networked ontology, for instance to create a specific ontology for transport by road or by sea or transport of bulk cargo like liquids. In case different transport modes meet, it is for the purpose of exchanging cargo like containers, i.e. containers are transhipped from a truck onto a vessel and the concept ‘container’ with all its related concepts and attributes is shared. Networked ontologies thus cater for all specifics of for instance each modality and combines them if these modalities meet. Note that the ontology does not model ‘business transaction’ and ‘interaction’ as these are of a more dynamic nature.

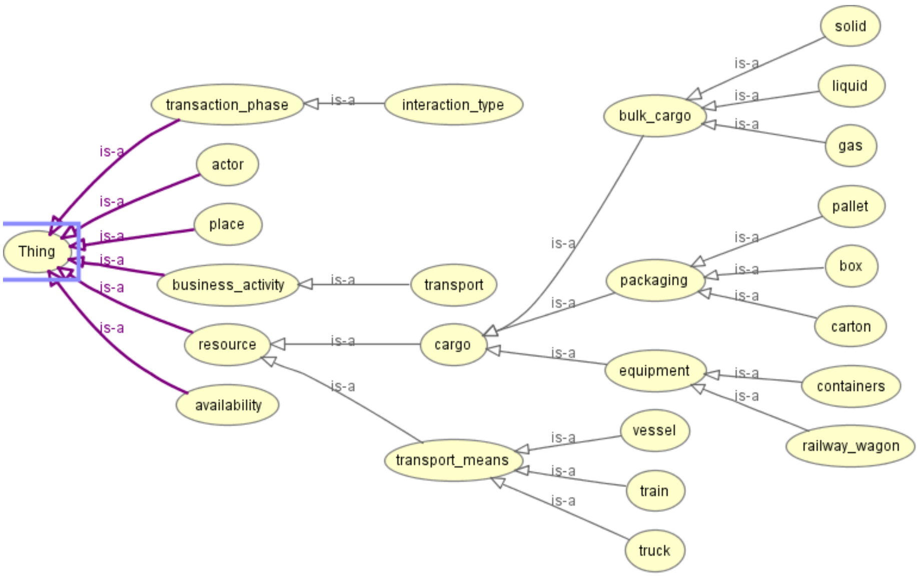


Fig. 2. Classes of transport interoperability ontology

A transport interoperability ontology is not only a specialization of concepts like ‘resource’ in ‘cargo’. There need other types of associations to be specified, e.g. the relation between ‘container’ and ‘packaging’. This type of association needs to state that a particular container with its measurements, e.g. a twenty feet container, can only contain a maximum number of pallets of a particular type (e.g. Europallets). Additionally, rules need to be added, e.g. specifying that liquid or gas bulk cargo needs a particular type of container for its transport and/or can be transported by pipeline. Thus, further semantics is added to the ontology.

3 Technology Support for Data Sharing and Tools

Whereas the previous section defines shared business concepts modelled with ontology that can be specialized to a specific application area, a technical syntax is required to actual share information as part of a business transaction that leads to value exchange according a value proposition, e.g. to actually transport cargo from one location to another. Information sharing can be implemented in different ways. The business interoperability ontology and its specializations need to be mapped or ‘grounded’ [15] to different architectural solutions. This section presents the architectural solutions that have to be supported and presents an upper ontology representing these solutions as a means for grounding business concepts. This uppor ontology can be used to manage architectural grounding. The ontology presented in this section is the combination of the architectural grounding and the business interoperability ontology and is called the interoperability ontology. Tools are briefly for specializing the business interoperability ontology and managing architectural grounding are briefly discussed in this section.

3.1 Technology Support of Business Concepts

To actually share information for value exchange, technical solutions are required. Traditionally, business documents were used, but gradually information technology is taking over. Messaging with EDI (Electronic Data Interchange, still applied widely in trade and logistics) or XML (eXtensible Markup Language) is still the most used paradigm, but there are other paradigms. As there is more than one paradigm, a selection has to be made, which is called 'architectural grounding' in this paper. Firstly, a distinction between asynchronous and synchronous communication is to be made, and secondly a supporting technology within the context of asynchronous or synchronous communication. Asynchronous communication implies that business processes of enterprises are independent of each other but need to synchronize data for certain activities. Synchronous communication means that a business process of a service providing enterprise needs to deliver a result before a business process of a consuming enterprise can continue. Synchronous communication is for instance required if an ETA (Estimated Time of Arrival) of a vessel is required for a container booking, but basically logistic business seems to be asynchronous. Figure 3 shows an overview of technology support for enterprise interoperability. Business interoperability consists of business activities with their value proposition and supporting semantic models (section 2.1).

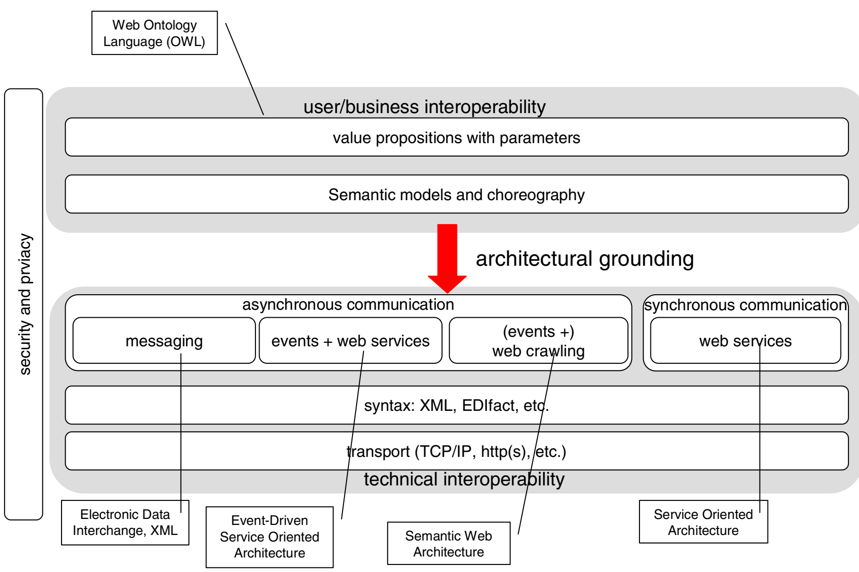


Fig. 3. Functionality for value webs

Asynchronous communication is supported by three paradigms, namely (1) messaging, (2) Event Driven Architecture combined with a Service Oriented Architecture (EDSOA), and (3) Semantic Web Architecture (SWA). An Event Driven Architecture (EDA) can also be combined with an SWA, where an event serves as a trigger to collect data [14]. Whenever an EDA is implemented, there has to be a mechanism for sharing business data, e.g. a web service to collect the data. A Service Oriented Architecture (SOA) supports synchronous communication. The objective of all these paradigms is to share data, e.g. by directly exchanging it (messaging) or capturing it based on crawling (SWA). The underlying technology for these paradigms will differ, e.g. messaging requires adapters that support transformation for data submission and SWA requires a web server for data publication and a crawler for data capture. These paradigms are supported by syntax and transportation protocols.

Each of these architectural solutions can be further decomposed into concepts shown in a technical interoperability ontology. EDI consists for instance of core components that construct EDI segments for messages (UNSMs: United Nations Standard Messages [12]). A web service ontology consist of ports, bindings, etc. like shown in for instance [22]. Figure 4 shows the meta model for EDI, XML, events, and web services. In XML, individual elements are not managed separately, but from a modelling view they can be. Therefore, they are distinguished. A particular UNSM or XML Schema Definition (XSD) is an instance of concepts shown in this figure. The figure shows that XSDs can be used as parameter in a web service and can also represent events. Complex structures like composite elements or composite web services are not depicted, they can be added easily.

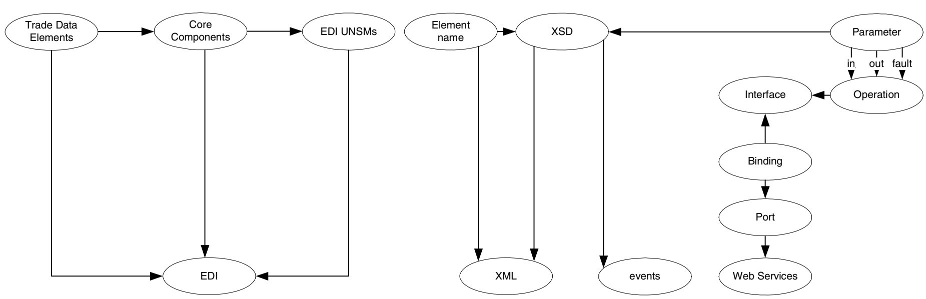


Fig. 4. Ontology for managing technical solutions

Most IT systems currently support messaging and web services; EDA is basically supported by ESBs with a publish-subscribe pattern that integrates different IT systems (hub-spoke IT solutions are mostly based on an ESB). ESBs can also be interconnected to construct so-called federate ESBs. An ESB can also support composite web services based on open standards for processes integrating web services of different IT systems (BPEL: Business Process Execution Language). Data Base Management Systems support SWA by making data available in RDF (Resource

Description Framework) or can be extended by open source add-ons like D2R (http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/).

3.2 Management: Interoperability Ontology

An interoperability ontology that combines the business and technical interoperability ontology can serve as a user interface for managing networked ontologies and their architectural grounding. Such an interoperability ontology should describe the relation between parts of the business and technical interoperability ontology like mapping to EDI, representation of concepts by their XML element names, or mapping to an existing XML Schema. The relation between the business and technical interoperability ontology can be constructed in different ways, e.g. an ontology representing instances shared by a particular interaction, a lower ontology representing for instance 'resources', or an ontology for data shared for a particular business activity, like 'transport', is mapped to an existing EDI UNSM or XSD.

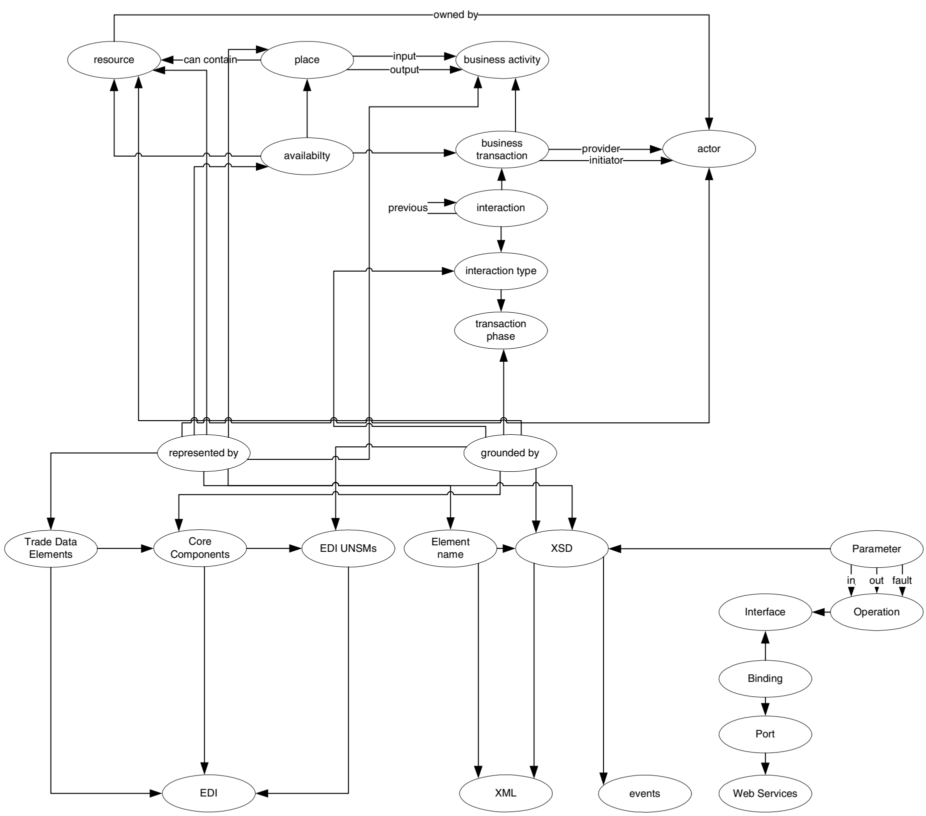


Fig. 5. Interoperability ontology for managing architectural grounding

Figure 5 shows two approaches to architectural grounding, namely representation of concepts in another syntax than OWL, e.g. by trade data elements, XML elements or groups like core components and XSDs, and grounding reflecting a mapping of ontologies to existing structures like UNSMs and XSDs and their implicit semantics. The next figure is not complete, since it does not distinguish between individual concepts and groups of concepts, e.g. ‘container’, represented in a syntax or mapped to existing structures. These groups can be used as parameters in a web services, but can also be re-used in an XSD reflecting the data that can be exchanged by for instance a particular interaction. For example, the concept ‘container’ and all its properties, modelled by a ‘container’ ontology, can be grounded to an XSD that is an output parameter of a web service requesting all containers loaded on a vessel. The same ontology and its grounding to EDI core components can also be applied in a transport interoperability ontology of a transport instruction that is grounded to an EDI UNSM like a Customs Declaration.

Grounding requires transformation rules. XSLT (eXtensible Stylesheet Language Transformation) is an example for defining such transformation rules. EDI requires the addition of specific constants in the transformation, i.e. the so-called ‘qualifier’, and selection of applicable value ranges of a data element, e.g. lengths are always given in metres with a maximum value.

3.3 Tools for Specializing the Business Interoperability Ontology

Tools for specializing the business interoperability ontology to an application are like logistics and supporting the architectural grounding to a syntax have to support the interoperability ontology (Figure 5) or parts of it. Basically, we identify two components (Figure 6): the Business Interoperability Modelling component to support the specialization of the business interoperability ontology to an application area and the Architectural Grounding Component to support architectural grounding to the technical interoperability ontology. The Business Interoperability Modelling Component has to deliver ontologies for one or more business activities, including their choreography for business transactions. These ontologies can be based on existing ontologies or data structures represented for instance by a UML class diagram. The Architectural Grounding Component has to provide the syntactical representation for interaction types (section 2.1), e.g. by an XSD or a web service. We will discuss how the functionality of these components is currently supported by tools.

Ontology tools like Protégé provide functionality for the Business Interoperability Component. They support re-use of other ontologies to construct networked ontologies, whether or not these ontologies are specified by other tools (at least if they adhere to an open standard like OWL, Web Ontology Language). They show which ontologies are imported to construct a new ontology. There are also extensions to ontology tools to transform UML class diagrams like the WCO data model version 3 [21] to an ontology (see for instance http://www.eclipse.org/m2m/atl/ usecases/ ODMImplementation/). These tools can also be used to operate with the business interoperability ontology as a basis for further specialization to an application area, but in general do not offer functionality to manage networked ontologies. Ontology management functionality would for instance show the relationship of networked ontologies and give warnings if one ontology is changed or deleted. Change and deletion may affect other ontologies. The business interoperability ontology can be the basis for managing networked ontologies for an application area. Quite a number of ontology tools support the creation of instances in RDF (Resource Description Framework) with for instance forms or import of XML data.

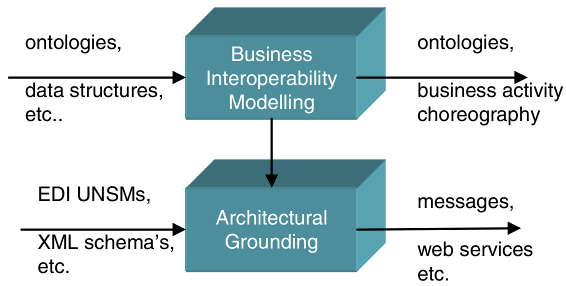


Fig. 6. Components for managing interoperability ontologies

Current tools most often have a function reflecting the 'represented by' associations for architectural grounding between the business and technical interoperability ontology. This function can be applied to generate for instance and XSD for (part of) an ontology. There is a limited set of tools that is able map ontologies to existing XML Schema's or EDI messages, e.g. Gefeg used by WCO to map the data model to existing EDI messages. As there are no standardized mapping rules of class diagrams or ontologies, each tool that supports those mappings will offer different functionality.

4 Conclusions and Further Research

The paper presents an interoperability ontology decomposed in a business interoperability ontology that is specialized to logistics and its architectural grounding to a syntax for information sharing. The business interoperability ontology is based on 'value propositions' for actual value exchange in organizational networks or value webs. The paper briefly discusses tools. With respect to the implementation, other issues are for further research. For instance, the ontology presented in this paper can be used as an intermediate for ontology matching between different technical solutions of two or more (closed) value webs. By importing the technical structures like XSDs and EDI implementation guides and deriving semantics out of these structures, this semantics can be matched with a intermediate ontology like presented in this paper. Matching can be the basis for constructing adapters between these value webs.

Instances of the concepts 'business transaction' and 'interaction' are more dynamic, i.e. there will be more and these will change more frequently than instances of other concepts. Further research is required as to how these two concepts can be the basis for developing new type of information systems or are implemented in existing IT solutions.

Other aspects for further research are the adaptation of existing tools to support the ontology with all its required functionality and further specify management functionality for networked ontologies based on the interoperability ontology presented in this paper. Currently, several research projects in logistics and supply chain security are further developing and applying the concepts and tools, but the approach will strengthen if it can also be applied to other areas.

Acknowledgement. This paper presents work in progress funded by the Dinalog Extended Single Window project and the EU FP7 SEC Cassandra project. These projects have to deliver a logistics interoperability ontology for global supply chain interoperability with its supporting tools.

References

1. Erl, T.: Service-Oriented Architecture – concepts, technology, and design. Prentice Hall (2005)
2. Spohrer, J., Kwam, S.K.: Service Science, Management, Engineering and Design (SSMED) – An emerging Discipline – Outline and references. International Journal on Information Systems in the Service Sector (May 2009)
3. European Interoperability Framework for Public Services (EIF), European Commission – IADBC (2009)
4. Overbeek, S., Klievink, B., Janssen, M.: A flexible, event-driven, Service-Oriented Architecture for orchestrating service delivery. IEEE Intelligent Systems (2009)
5. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web, a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American (2001)
6. Fast, Book of Search, Fast Data Search (2006)
7. Verberne, S., Van Halteren, H., Theijssen, D., Raaijmakers, S., Boves, L.: Evaluating Machine Learning Techniques for Re-ranking QA Data. In: Proceedings Workshop Learning to Rank for Information Retrieval (2009)
8. Hiemstra, D.: Using language models for information retrieval, PhD Thesis, University of Twente (2001)
9. Berre, A., Elvesæter, B., Figay, N., Guglielmina, C., Johnsen, S., Karlsen, D., Knothe, T., Lippe, S.: The ATHENA Interoperability Framework. In: Goncalves, R., et al. (eds.) Enterprise Interoperability II, pp. 569–580. Springer, Heidelberg (2007)
10. Vujasinovic, M., Ivezic, N., Kulvatunyou, B., Barkmeyer, E., Missikoff, M., Taglino, F., Marjanovic, Z., Miletic, I.: Semantic mediation for standard-based B2B interoperabilitity. IEEE Internet Computing 14(1) (2010)
11. OWL-S, Semantic Markup for Web Services, W3C member submission (2004)
12. Hofman, W.J.: EDI, Web Services and ebXML – interoperability in organisational networks, Uitgeverij Tutein-Nolthenius (2003) (in Dutch)
13. Heath, T., Bizer, C.: Linked Data – evolving the Web into a Global Data Space. In: Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers (2011)
14. Hofman, W.J.: Applying semantic web technology for interoperability in freight logistics. In: eFreight 2011 (2011)
15. Fensel, D., Kerrigan, M., Zaremba, M. (eds.): Implementing Semantic Web Services – the SESA framework. Springer, Heidelberg (2008)

16. Kim, J.: Supervenience and Mind: Selected Philosophical Essays. Cambridge University Press, New York (1993)
17. Bratman, M.E.: Faces of Intention. Cambridge University Press, Cambridge (1999)
18. Hruby, P.: Model-Driven Design using Business Patterns. Springer, Heidelberg (2006)
19. Business Process Modelling Notation (BPMN), version 1.2 (January 2009)
20. Hofman, W., van Staalduinen, M.: Dynamic Public Service Mediation. In: Wimmer, M.A., Chappelet, J.-L., Janssen, M., Scholl, H.J. (eds.) EGOV 2010. LNCS, vol. 6228, pp. 327–338. Springer, Heidelberg (2010)
21. World Customs Organization Data Model (version 3) – cross border transactions on the fast track (brief introduction), WCO (2009)
22. Elyacoubi, N.E., Belouadha, F.-Z., Roudies, O.: A metamodel of WSDL Web services using SAWSDL semantic annotations. In: IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2009, May 10-13, pp. 653–659 (2009)
23. Dietz, J.L.G.: Enterprise Ontology, Theory and methodology. Springer, Heidelberg (2006)
24. Börger, E.: High Level System Design and Analysis Using Abstract State Machines. In: Proceedings of the International Workshop on Current Trends in Applied Formal Method: Applied Formal Methods, pp. 1–43 (1998)

ICSP 2011 PC Co-chairs' Message

Cloud computing, service-oriented architecture, business process modelling, enterprise architecture, enterprise integration, semantic interoperability—what is an enterprise systems administrator to do with the constant stream of industry hype surrounding him, constantly bathing him with (apparently) new ideas and new "technologies"? It is nearly impossible, and the academic literature does not help solve the problem, with hyped "technologies" catching on in the academic world just as easily as the industrial world. The most unfortunate thing is that these technologies are actually useful, and the press hype only hides that value.

What the enterprise information manager really cares about is integrated, interoperable infrastructures that support cooperative information systems, so he can deliver valuable information to management in time to make correct decisions about the use and delivery of enterprise resources, whether those are raw materials for manufacturing, people to carry out key business processes, or the management of shipping choices for correct delivery to customers. Cooperative information systems provide enterprises and communities of computing application users with flexible, scalable and intelligent services in large-scale networking environments typical of industrial enterprises.

The OTM conference series has established itself as a major international forum for exchanging ideas and results on scientific research for practitioners in fields such as computer-supported cooperative work (CSCW), middleware, Internet/Web data management, electronic commerce, workflow management, knowledge flow, agent technologies and software architectures, to name a few. The recent popularity and interest in service-oriented architectures and domains require capabilities for on-demand composition of services. Furthermore, cloud computing environments are becoming more prevalent, in which information technology resources must be configured to meet user-driven needs. These emerging technologies represent a significant need for highly interoperable systems. As a part of OnTheMove 2011, the Industry Case Studies Program emphasized research/industry cooperation in these future trends. The focus of the program was on the discussion of ideas where research areas address interoperable information systems and infrastructure. Industry leaders, standardization initiatives, European and international project consortiums were invited to submit position papers discussing how projects within their organizations addressed software, systems and architecture interoperability.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 136–137, 2011.
© Springer-Verlag Berlin Heidelberg 2011

An international Programme Committee reviewed proposals and we present here strong papers presenting the implementation and use of current integration technologies in the implementation of real enterprise cooperative information systems. We hope that you find this industry-focused part of the program valuable as feedback from industry practitioners, and we thank the authors for the time and effort taken to contribute to the program.

August 2011

Hervé Panetto
Richard Mark Soley

Realising the Flanders Research Information Space

Peter Spyns and Geert Van Grootel*

Vlaamse overheid – Departement Economie, Wetenschap en Innovatie
Koning Albert II-laan 35, bus 10, B-1030 Brussel, Belgium
{Peter.Spyns,Geert.VanGrootel}@ewi.vlaanderen.be

Abstract. A model driven architecture combined with SBVR and fact oriented modelling are key standards and methodologies used to implement a Flemish research information government portal. The portal will in the future offer various services to researchers, research related organistions and policy makers.

1 Context

The knowledge economy is one of the cornerstones of our modern society. Hence, it is very important for a country or region to have a good overview of its science and technology base to develop an appropriate policy mix of measures to support and stimulate research and innovation. Needing a new system for managing research information, The Flemish department of Economy, Science and Innovation (EWI) launched the Flanders Research Information Space programme (FRIS) [9]. The FRIS concept creates a virtual research information space covering all Flemish players in the field of economy, science and innovation. Within the space, research information can be stored and exchanged in a transparent and automated way. A first realisation is the FRIS research portal (www.researchportal.be) to provide up to date research information on projects, researchers and organisations in Flanders. One can browse for projects, individual researchers, research groups and collaborations. Also other European countries, which are member of the European Organisation for International Research Information[1], have deployed similar initiatives.

A key feature is that data can be immediately collected at the point of creation in the operational processes of data providers (e.g., universities, funding bodies …). The data are up-to-date and are supposedly more accurate. Also, parallel data gathering processes can be eliminated, resulting in a lot of administrative work being spared. As many data providers are involved (universities, funding agencies, independent research institutes to only name the most important ones) it is not a surprise that the FRIS portal team is confronted with various (external) legacy systems, data heterogenity, different classifications, different storage formats, data of varying quality, different data input and upload techniques and so on.

Having a standard E-R schema for the research domain, c.q. the CERIF model [3][2], as interchange format is not enough to guarantee flawless interoperability and

* In alphabetical order.

[1] www.eurocris.org

[2] The Common European Research Information Format (CERIF) is a recommendation to EU member states for the storage and exchange of current research information.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 138–141, 2011.
© Springer-Verlag Berlin Heidelberg 2011

information exchange with other (external) applications and services. Agreement (or least univocity) should be achieved on the terminology of the CERIF schema.

2 FRIS High Level Engineering Principles

To realise a "robust" FRIS, the technical solution aimed at is to build a "modelling process pipe line". Using a conceptual modelling method results in a collaboratively and agreed upon conceptual model that would automatically be converted in a CERIF-based E-R model that, in its turn, is implemented as optimised MySQL database tables. This corresponds to a model driven architecture. Mechanisms to transform business rules from a high level conceptual representation into efficiently executable low level software modules are needed and must be very robust to frequent updates of the business rules.

Domain experts, not familiar with specific database modelling methods, should be able to adapt and extend the conceptual domain model without any further intervention on the "lower" levels. Therefore, EWI has decided to express the business facts in the domain concerned in an SBVR-style (Semantics of Business Vocabulary and Business Rules [5]). From the SVBR specification document [5: pp. 371–382] follows that fact oriented modelling methodologies – e.g., Object Role Modelling (ORM) [2] – can help in freeing domain experts from having to become familiarised with arcane and technical (at least to them) formalisms. Expressing the semantics of business rules in an easily understandable way for domain experts is paramount. Some even consider agreeing on the business rules more difficult than agreeing on the domain concepts and relationships [4].

3 Two Cases

3.1 Business Vocabulary Definition

To ground the vocabulary of the FRIS research portal, a concept definition storage is used. It provides access to the meaning of the concepts used by the FRIS. In the spirit of SBVR this is achieved by means of semantic communities (i.e. a group of people communicating on the same topic), which are subsequently divided into linguistic communities (according to the language the various subgroups of a semantic community use). The various communities can exchange meaning descriptions of concepts, map them on concept labels and determine "rule sets".

The SBVR standard of the Object Management Group (OMG) is said to be unique in its completeness and flexibility as it unifies several strong points:

- a combination of some ISO standards on terminology and vocabulary (e.g., ISO 37, 704, 1087);
- support for formal logics as basis via fact-oriented modelling;
- linguistics and linguistic annotation of grammar for natural language, including multilinguality;
- practical experience by application in large, complex organisations.

In practice, the FRIS requirements imply not only semantically reconciliating several science classification schemes and lists of scientific topic codes but also more general ISO code tables on countries, languages, currencies as well as local Flemish standard etc. Hence, a list of concepts described as unambiguously as possible is required (together with a tool to manipulate and maintain this list). E.g., the science classification (IWETO) used by the Flemish fundamental research funding agency has been mapped into a classification of core technologies (based on the classification of Fraunhofer Institute of Systems and Innovation Research) used by the Flemish innovation funding agency. Thanks to this mapping industry patents can be related to research projects – e.g. to implement a service that can determine the impact of research projects.

To set up the “semantic machinary” behind the FRIS, EWI has contracted an external consultant[3] that has developped its own business semantics management methodology [1] and corresponding development environment based on a scientific fact oriented ontology engineering methodology and tools [6].

3.2 Grounding Business Rules

The above mentioned efforts address the problem of diverging interpretations of the business rules vocabulary [7,8]. A common conceptual model for the CERIF E-R schema and established a consensus amongst the involved parties (here: Flemish university data centres’ information analysts). Concepts of the semantic dictionary are combined to create semantic patterns (corresponding to binary fact types), which are then semantically restricted by constraints expressing specific business rules. These rules, i.e. specific constraints on the relationships, are kept separated from the semantics of the domain concepts and relationships. The CERIF concepts can be re-used. Its meaning stays the same even if the nature of the relation between the vocabulary changes in other rules for the same application (or even for other applications).

4 Discussion

The stakeholders are positive regarding the approach and way of working as they recognise the advantages. In addition, the stakeholders appreciate the apparent promise on scalability and better maintainability of the data and knowledge in the FRIS research portal. Experiments have shown that the above described way of working can result in a substantial time and hence cost reduction [7]. However, many hurdles – in particular data quality control at the input source – still need to be overcome.

5 Future Work and Conclusion

The FRIS portal is up and running for two years. In the future, the FRIS research portal will roll out a range of new services as part of the research information space. The possibilities are numerous: a white guide (who does what?), a library of publications by a particular researcher (digital library), a service for updating and reusing researchers’ CVs and provision of information on patents, to name but a few. It is

---

[3] Collibra NV.

expected that a model driven architecture and applying a fact oriented modelling methodology to define SBVR-style business rules and establish a semantic domain dictionary prove to be the right methods to realise these services. The cases presented in this paper illustrate two important steps towards this end. But it surely remains an ambitious, long term challenging endeavour and, for the Flemish government, it is an innovating[4] and fresh [‘fris’ = ‘fresh’ in Dutch] approach.

References

1. De Leenheer, P., Christiaens, S., Meersman, R.: Business Semantics Management with DOGMA-MESS: a Case Study for Competency-centric HRM. Journal of Computers in Industry 61(8), 760–775 (2010)
2. Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann, San Francisco (2008)
3. Jörg, B.: The Common European Research Information Format Model (CERIF). Data Science Journal 9, 24–31 (2010); Jeffery, K., Asserson, A. (eds.)
4. Meersman, R.: Ontologies and Databases: More than a Fleeting Resemblance. In: d’Atri, A., Missikoff, M. (eds.) OES/SEO 2001 Rome Workshop. Luiss Publications (2001)
5. Object Management Group, Semantics of Business Vocabulary and Business Rules (SBVR) v1.0, http://www.omg.org/spec/SBVR
6. Spyns, P., Tang, Y., Meersman, R.: An ontology engineering methodology for DOGMA. Journal of Applied Ontology 1-2(3), 13–39 (2008)
7. Spyns, P., Van Grootel, G., Jörg, B., Christiaens, S.: Realising a Flemish govern-ment innovation information portal with Business Semantics Management. In: Proceedings of CRIS 2010, pp. 45–53 (2010)
8. Spyns, P., Van Grootel, G.: Re-engineering Business Rules for a Government Innovation Information Portal. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 565–574. Springer, Heidelberg (2010)
9. Van Grootel, G., Spyns, P., Christiaens, S., Jörg, B.: Business Semantics Management Supports Government Innovation Information Portal. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 757–766. Springer, Heidelberg (2009)

[4] In 2009, the FRIS received the innovation prize by the Flemish Government.

Discovering Potential Synergies between Research Projects and Standardisation, the COIN Case

Arianna Brutti[1], Piero De Sabbata[1], Nicola Gessa[1],
Cristiano Novelli[2], and Enrico Del Grosso[3]

[1] ENEA UTT-PMI, Via Martiri di Monte Sole. 4, 40129, Bologna, Italy
{arianna.brutti,piero.desabbata,nicola.gessa}@enea.it
[2] FTI, Forum per la Tecnologia dell'Informazione, Piazzale Morandi. 2, 20121 Milano, Italy
cristiano.novelli@forumti.it
[3] TXT eSolution, via Frigia 27, 20126, Milano, Italy
enrico.delgrosso@txt.it

Abstract. The paper aims to present a methodology developed to support the COIN IP project in identifying potential synergies between the project outcomes and the world of standards and standardisation. The major references for our purposes are the COPRAS project outcomes and its guidelines to standardisation. Nevertheless, in order to apply such approach to the COIN concrete case, it has been necessary to setup a methodology able to reduce the complexity of the domains under investigation and to extract a richer information. The paper presents also some aspects and outcomes of its application on COIN.

Keywords: Standard, web services, enterprise interoperability, enterprise collaboration.

1 Introduction

This paper presents an in progress work about the investigation of the potential synergies between an IT research project and the world of standardization that has been committed by COIN project (www.coin-ip.eu).

COIN is an integrated research project in the European Commission Seventh Framework Programme (EU FP7) that is focused on technologies and business models related to Enterprise Collaboration and Interoperability.

The aim of COIN is to prepare the methodologies, the technologies and the business models that allow the "COIN vision by 2020" to come true [1].

On the purpose to disseminate the project outcomes a community has been created, the COIN Angels, aiming to play a more active role of "ambassadors and evangelists" of the COIN vision and results. In this framework the activity reported in this paper has been launched with the aim to collect, analyse and frame all the potential implications between standardisation issues and COIN Project outcomes; we present our methodology in paragraph 3 and its application in the context of COIN project in paragraph 4.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 142–147, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 The Starting Point

The COIN commitment has been to collect, analyse and frame all the potential implications between standardisation issues and COIN project outcomes.

The request was tackled from two different perspectives:

- how the project outcomes could contribute to the standardisation
- how the project outcomes (and actors) can get advantage from standards.

The main reference about facilitating the relationships between an IT research project and standardisation are the outcomes of the COPRAS (COoperation Platform for Research And Standards, http://www.w3.org/2004/copras/) project.

Started in 2004, it was aiming to improve the interfacing, cooperation and exchange between IST (Information Society Technologies) research projects and ICT standardization. Partners of the project were the three European standardization bodies (CEN, CENELEC and ETSI), the W3C Consortium (W3C) and The Open Group (TOG).

The objective for COPRAS was tackled mainly by supporting projects in establishing relationships with the main standards organizations.

As one of its most relevant deliverables, COPRAS in 2007 published a set of Generic Guidelines [2] for IT projects.

The approach of the guidelines could be summarized on three corner stones:

- standardization as an opportunity for IT projects to disseminate their outcomes
- identification and planning of standardization established since the beginning of the project activities
- direct involvement of the project with standardization bodies.

COPRAS has been our reference but it has required some additional effort in order to be applicable in our context, due to some aspects of the COIN case:

- the project, being a large integrated project, has produced a great amount and variety of outcomes, in a wide range of domains, unease to manage by a single expert;
- the time frame of the project do not fit the time scale of a standardization initiative and, consequently, the project itself is not interested in being an actor of standardization in first hand;
- the project has also a ‘visionary’ course being connected with the ideas of the future evolutions of the Internet;
- it is important to prioritize project standardization potentialities with a reference to the European future standardization policies;
- the project is interested to think in a bidirectional way the relationship with the standardization world.

For these reasons the approach to be implemented has to fulfill the following additional requirements:

- subdivide the domain of interest, to facilitate the reconciliation between project outcomes and standardization initiatives in the same sub-domain

- deal with skills of different people highly specialized
- consider the ‘political’ perspective (and priorities) of standardization world
- consider the relationships between the project and standardization as bidirectional
- implement different types of relationships between project and standardization.

The result of the processing of these requirement has been the methodology that is presented in the following paragraph: in short it moves from a mono dimensional investigation (*how my results could be standardized*) to a three dimensional evaluation (*potential of my outcomes for future standardization, potential of present standardization for my future developments, potential interest for European standardization policies and priorities*).

3 The Methodology

The proposed methodology is a process that can be applied in order to identify potential synergies between research projects outcomes and the world of standards and standardisation.

Talking about identifying synergies we intend checking *if* and *how* the results achieved within one world could contribute in improving the other one.

The methodology goes through the following steps:

1. **definition of a domain grid** in order to subdivide the domain under analysis in respect to the topics/problems on which we want to compare and to reconcile the world of standardization (existing or coming specifications) and the project outcomes;
2. **collection of existing standards or running standardisation initiatives** in order to select the relevant ones with reference to each of the sub-domains defined by the grid;
3. **analysis and selection of project outcomes** in order to identify their reference in the grid and evaluate their potential relationships with actual or future standards; the evaluation should be made taking into account two factors: *innovation* (which represents the degree of innovation of the outcomes in respect of the status of art) and *generality* (which represents the measurement of the applicability of the project outcomes to a wider domain);
4. **matching between promising project outcomes and standardisation initiatives and policies** in order to understand if they could become or contribute to a new standard, provide inputs to existing standards or simply represent a request for new standards;
5. **sharing of the conclusion with project and standardisation stakeholders** in order to agree on potential development and possible actions.

4 Construction of the Domain Grid

In this section we focus on the description of the domain grid for standard analysis and categorisation.

In order to define a classification of the domain of the standards that could be related to eBusiness, Interoperability and more specifically to Interoperability Service Utilities (ISU [3]) in the COIN project perspective we examined different approaches[4]: based on a classification of the standards for eBusiness [5], on levels of interoperability [6], [7], [8] (for an overview see also [11]), on building elements of an Interoperability Framework [9].

The latter approach resulted more suitable as a starting point to partition our domain and to highlight both the standardization bricks of the interoperability construction and the components of the COIN system. To emphasize issues related to organization level and common specification customization (like the case of UBL and related specializations[10]) we adopted an hybrid approach that produced the following grid.

The first column identifies the three different levels of interoperability: technical, semantic and organizational (as defined in [6]).

The second column identifies the categories of artifacts that are enablers for each interoperability level:

- **Technical interoperability:** the enablers are related to technologies for communication, messaging mechanisms and syntax for artifact building.
- **Semantic interoperability:** it is worth to observe that the first enabler group is often the support for the others (for example Schematron is used for defining UBL profiles), and, the same could for the second one supporting 'profile based specifications (in the table expressed by thin arrows).

  The groups resulting from this categorization are relatively homogeneous for quality of the semantic content, skills necessary for their developments, standardization organizations or consortia in charge for their development, dependency on the applicative domain.
- **Organisational interoperability:** analogously there are two groups of enablers, "Domain independent collaboration description languages" and "Domain related specifications" that can use the first ones to express the specifications.

The third column reports samples of standardized specifications belonging to each of these groups.

The fourth column reports outcomes of the project having a possible relationship with the standardisation belonging to the same subdomain, like, for example:

- the result could contribute to further development of an existing specification;
- the result could become the basis for a new specification;
- the result could take advantage from an existing specification
- the result has been hampered by the lack of specifications.

The actual grid for COIN is in figure 1.

The application of the first three steps of the methodology to a first group of COIN outcomes allowed to identify 17 of them with a potential relationship with standardisation and to attribute them a score value (range 0-3) in terms of Innovation and Generality.

The selection of the items beyond the threshold of '2' for both the indicators leaded to choose 5 of them with a promising relationship with standardisation (in figure 1, on the right column there are two samples).

Interoperability level	Enablers	Standard samples	Collected outcomes
ORGANISATION	Domain related specifications	Insurance: ACORD Textile Clothing Footwear reference: eBIZ-TCF Public Procurement reference: WS-BII ...	
	Domain independent specifications to achieve collaborations and retrieve resources	Business Process and Collaboration Descriptions (ebXML ebBP and CPPA, BPMN, WS-BPEL, WS-BusinessActivity, ...) Collaboration Establishing and Resources Retrieval (ebXML Registry, UDDI, BCM, WS-Discovery, WS-MakeConnection, SPML, ...) ...	"**Trusted Information Sharing (TIS)**" service. *It could provide a base model for trust and accounting in document-centric information sharing.*
SEMANTICS (Data models, Dictionaries, Ontologies,...)	Vertical information representation and exchange specifications (per sector or per domain)	Native: Insurance: ACORD Health care: HL7, LOINC	
		Profile Based: Textile clothing Footwear: UBL profile for eBIZ-TCF (sector based) Public Procurement: UBL profile for WS-BII (domain based)	
	Horizontal information representation and exchange specifications (to be specialized)	UBL, WORDNET, ...	"**Semantic Cluster Management Services**", based on innovative product structure ontology. *It could suggest improvements or new ontologies for managing product structure*
	Semantic description languages	RDF, OWL, SCHEMATRON, CAM, XRD, WS-Context, WS-SecurityPolicy, XLIFF , ...	
TECHNICAL	Messaging Mechanism (Web Services)	Service Composition (WSBPEL,...)	
		Service Discovery (UDDI, ...)	
		Service Description (WSDL, ...)	
		Digital Envelope (SOAP, ebMS, ...) Services Coordination (WS-BPEL, WSN, WS-Transaction, WS-AtomicTransaction WS-Coordination, ...) Services Security and Reliability (WS-Security, WS-SecureConversation, WS-ReliableMessaging, WS-Trust, ...)	
	Syntax	XML Technologies (XML, XSD, ...)	
	Communication Mechanism (Internet)	Transport (HTTP, SMTP, FTP, BEEP, ...)	n/a
		Networking (TCP/IP)	n/a

Fig. 1. The domain grid

5 Conclusion

The methodology we propose in this paper to investigate the potential synergies between large IT research projects and standardisation is a work in progress. It is formalised in five steps and presently the first three ones have been proved on the COIN IP project; the remaing ones are expected to be experimented in a near future.

The current status of the work demonstrates that it is possible to systematically identify the potential synergies between the standardisation world and the project despite a) the complexity of the project activities and outcomes, b) the absence of experts having a strong background on both worlds in the same person. The methodology has allowed different experts to combine their respective expertises and visions to tackle the problem in a structured way.

The next developments, after the completion of the test of the whole methodology, will be a) to better define the roles played by the different expertises –as individuals, and organisations- in each step, b) to test its generality towards other cases.

References

1. COIN Integrated Project, `http://www.coin-ip.eu`
2. Standardization guidelines for IST research projects interfacing with ICT standards organizations, COPRAS project (January 2007), `http://www.w3.org/2004/copras/docu/D27.pdf`
3. Li, M., Cabral, R., Doumeingts, G., Popplewell, K.: Enterprise Interoperability Research Roadmap, final version 4.0, Information Society Technologies (July 2006), `ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/ebusiness/ei-roadmap-final_en.pdf`
4. Folmer, E., Verhoosel, J.: State of art on semantic Information Systems Standardisation, interoperability and quality, TNO, Universiteit Twente, NOiV, CTIT (March 2011) ISBN 978-90-9026030-3
5. Blind, K.: A Taxonomy of Service Standards and a Modification for E-Business. In: Jakobs, K. (ed.) Information Communication Technology Standardization for E-Business Sectors: Integrating Supply and Demand Factors, pp. 24–30. Information Science Reference (2009) ISBN 1605663204
6. EIF 1.0, European Interoperability Framework for pan-European eGovernment Services IDABC project, Belgium, European Communities (2004) ISBN 92-894-8389-X, `http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529`
7. EIF 2.0, Annex II - EIF (European Interoperability Framework). In: Communication Towards interoperability for European public services, December 16, European Commission (2010)
8. Kubicek, H., Cimander, R.: Three dimensions of organizational interoperability. European Journal of ePractice (January 6, 2009), `http://www.epractice.eu/files/6.1.pdf`
9. Hemant, J., Huimin, Z.: A Conceptual model for comparative analysis of standardization of vertical industry languages. In: MISQ Special Issue Workshop Standard Making: A Critical Research Frontier for Information Systems (2003)
10. Brutti, A., Cerminara, V., Cristiani, S., De Sabbata, P., Gessa, N.: Use Profile Management for B2B Standard Adoption: The Case of eBIZ-TCF. In: Cunningham, P., Cunningham, M. (eds.) Proceedings of e-Challenges 2009 Conference, IIMC International Information Management Corporation Ltd. IOS Press, Dublin (2009) ISBN 978-1-905824-13-7
11. Panetto, H.: Towards a Classification Framework for Interoperability of Enterprise Applications. International Journal of Computer Integrated Manufacturing 20(8) (2007)

Applications of Ontologies for Assembling Simulation Models of Industrial Systems

Petr Novák[1,2] and Radek Šindelář[1]

[1] Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems, Vienna University of Technology, A-1040 Vienna, Austria
{novak,sindelar}@ifs.tuwien.ac.at
[2] Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, 121 35 Prague, Czech Republic

Abstract. Simulation models are important parts of industrial system analysis and control system design. Despite the wide range of possible usage, their formalization, integration and design have not been satisfactorily solved. This paper contributes to the design phase of simulation models for large-scale industrial systems, consisting of a large number of heterogeneous components. Nowadays, it is the simulation expert who assembles the simulation model for a particular industrial plant manually, which is time-consuming and error-prone. We propose to use a semantic engine performing SPARQL queries, which assembles the structure of a simulation model semi-automatically, using formalized knowledge about real industrial plant and available simulation blocks represented in appropriate ontologies. As each real plant device can be represented by more than one simulation blocks, the selection of suitable simulation candidates is based on matching interfaces of neighboring blocks. Evaluation on a real-life industrial use-case shows improvements in reducing development time and avoiding design errors. Major results of this paper are the proposed structures of the ontologies and the SPARQL query realizing the selection of the appropriate simulation blocks.

Keywords: Ontology, simulation models, industrial automation, SPARQL querying, design and integration.

1 Introduction

Modern industrial automation systems are sophisticated systems comprising various computer algorithms and tools. Although the production automation has been investigated for several decades, the integration of diverse tools and sharing knowledge between particular engineering disciplines [8] still remain challenging problems. This paper is focused on semantic design of simulation models, being the software approximations of the behavior of real industrial systems.

Simulation models are able to support a lot of industrial engineering tasks. They can perform experiments that would be, for example, dangerous on the real plants themselves, such as testing diverse critical scenarios of nuclear power

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 148–157, 2011.
© Springer-Verlag Berlin Heidelberg 2011

plants [9]. Since some real experiments cannot be repeated under the same conditions, such as testing performance of buildings and their control algorithms as the outside weather is every time unique, the models can contribute to optimize the operation of such systems, too. Although simulation models can be used in many domains, their design and integration have not been satisfactorily solved.

The goal of this paper is to introduce a significant impact of using ontology-based methods on simulation model design phase. The proposed and current approaches are compared in Fig. 1. Instead of manual and error-prone work, we propose to formalize plant and simulation knowledge and to use a semantic engine assembling the structure of a simulation model automatically. The labels of the edges denote the kinds of data that flow between the engineering steps.

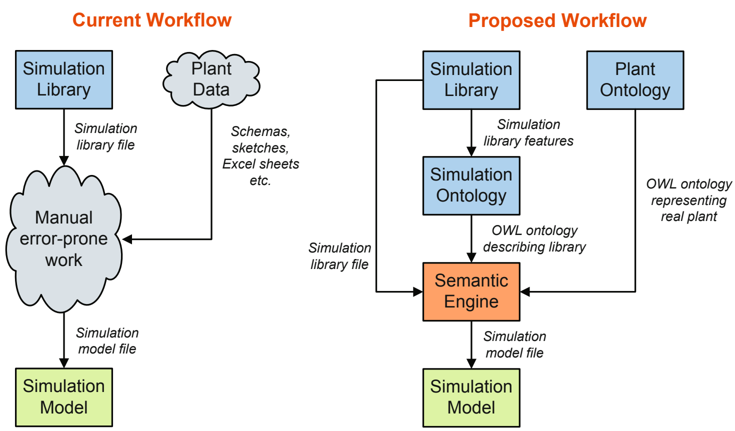


Fig. 1. Comparison of current and proposed workflows

The ontology-based formalization provides powerful support for various sound methods such as ontology querying and reasoning. It is efficient as heterogeneous data models occur on the simulation block level as well as automation system level. This methodology guarantees avoiding structural errors in simulation models design and as the meaning of signals is expressed, the semantic engine ensures a compatibility of interacting signals or detects errors in existing models, including e.g. different scales of the variables. On the other hand, our approach brings a requirement on a consistency of the describing ontologies. The presented methodology is limited for single-input and single-output blocks connected in series, but it can be generalized for an arbitrary kind of blocks.

The remainder of this paper is structured as follows: Section 2 summarizes the related work and Section 3 defines a problem scope and research issues. Section 4 describes our solution to formalize plant and simulation knowledge and the fifth section introduces ontology-based methods to support simulation model design. In the sixth section, the methodology is demonstrated on a use-case electrical circuit. The seventh section concludes and proposes further work.

2 Related Work

The term ontology originates from philosophy, where it refers to the theory of existence. In technical or scientific sense it is defined in many ways [1], [2]. One of the most cited definition is by Gruber: *"An ontology is an explicit specification of a conceptualization"* [3]. The core feature of ontologies is representing and sharing domain knowledge in a machine-understandable form. Ontologies comprise concepts (also called ontology classes), individuals (i.e. instances of ontology classes), properties and constraints. At a concept (class) level, the relations are usually declared and they are defined at an individual level.

In our approach, OWL DL[1] is used to represent the ontology. It was selected as a optimal compromise between expressive power and possibilities to perform reasoning in future work. We derive required information by SPARQL[2] queries; nowadays a reasoner is not used. For modifying or visualizing ontologies, we use Protégé[3], and the automatic semantic engine was implemented in Java, using ARQ[4], that is a query engine extending Jena Ontology API[5].

The Semantic Web technologies has been already used for simulation modeling [7]. In [4], the Ontology-driven Simulation Tool (ODS) is described. It is based on two ontologies that are mapped: a domain ontology categorizing a knowledge including a problem vocabulary, and a modeling ontology being used for the simulation model description. Our approach is based on a similar idea, but it addresses other engineering tools as well and reflects features of large-scale industrial systems.

In our approach, ontologies are used as a knowledge representation on a middle-ware layer that manages diverse engineering tools or simulation models of diverse simulators in a uniform way, in order to overcome one of the most important problems of simulation model design and integration: *"There is no commonly agreed upon format for representing and storing models"* [4]. The usage of ontologies for production automation is described, e.g. in [5], where can be found approaches dealing with further parts of automation system.

The presented task is an application of semantic integration [6] - i.e. the level of the system integration dealing with the meaning of exchanged data. In our case, the meaning of plant devices, simulation blocks, their interfaces and interchanged signals has to be stored. This request is satisfied in the automation ontology that is described in the following text. The presented approaches and algorithms were tested on simulation models implemented in MATLAB-Simulink[6], but they are not limited for this tool.

[1] http://www.w3.org/TR/owl-features/
[2] http://www.w3.org/TR/rdf-sparql-query/
[3] http://protege.stanford.edu/
[4] http://jena.sourceforge.net/ARQ/
[5] http://jena.sourceforge.net/ontology/
[6] http://www.mathworks.com/products/simulink/

3 Problem Scope and Research Issues

As the industrial plants typically contain devices that are repeated several times in their topological structure, one of the first tasks for the simulation expert is to analyze which devices will be simulated in the same way and what parameters are required. For example, a water distribution system involves devices such as piston pumps and rotary pumps, tanks, and pipes. The engineer has to decide, if the rotary pumps and piston pumps will be simulated by the same simulation block or if their behavior is so different that each kind of pump will be simulated separately. The functionality of the devices is comprised in so-called generic simulation blocks, i.e. blocks that have to be parameterized to obtain the simulation for the particular device. The parameters are, for example, diameters or lengths. The blocks are comprised in a so-called simulation library, also called universal simulation library to emphasize the necessity for parametrization.

A simulation model for the particular plant is created by copying generic blocks from the library into a simulation model file, interconnecting them and entering their parameters. A simplified example of the simulation library is depicted on the right side of Fig. 5. This library comprises generic simulation blocks representing an ideal source of voltage, ideal capacitor, and two resistors. Only the dominant inputs and outputs of the blocks are depicted; additional voltage inputs and outputs that are needed for implementation are omitted.

The research work presented in this paper was classified into following research issues.

RI-1: Formalization of industrial plant structure. The plant description is not usually well-structured, even it is not stored in an electronic form in many cases in industrial praxis. The formalization of the knowledge in a machine-understandable way, enabling to use automatic methods for retrieving required information and deriving new facts, improves design and runtime phases of automation systems.

RI-2: Formalization of universal simulation library features. Although simulation models are computer structures, a lot of pieces of information that was used during analysis and design phases of universal simulation libraries are not stored in a machine-understandable way. In contrast, it is usually given in some documentation and modifying the particular simulation model requires the simulation expert to study the rules for building the simulation model, the description of inputs and outputs, the meaning of signals etc.

RI-3: Industrial plant – simulation mapping. As the simulation models approximate the industrial plants and the generic blocks approximate the industrial devices, for both design and interpretation of simulation models the relationship between real industrial domain and the simulation domain is crucial.

RI-4: Semi-automated semantic assembling of simulation models. Assembling a simulation model for a particular industrial plant is a time -consuming and error-prone engineering task. For real large-scale systems, the simulation expert has to copy hundreds of blocks and enter thousands of interacting signals according to the real plant structure, as well as setting block names, parameters, and model interfaces.

4 Ontology-Based Formalization of Knowledge

This section summarizes our solution dealing with the formalization of plant and simulation knowledge and emphasizes their relationships. It addresses the research issues RI-1, RI-2, and RI-3.

Our approach is based on strong distinguishing between plant and simulation knowledge. We use a so-called plant ontology (RI-1) to represent the knowledge about the real plant, hence it comprises especially structure of the real industrial plant (i.e. particular devices, their interconnections etc.) and parameters of the plant (e.g. lengths or diameters). Since we can use more than one simulation models, the knowledge related to available simulation blocks is comprised in the simulation ontology (RI-2). This ontology represents information about the structure of the universal simulation library and its generic blocks. Important knowledge relates to inputs and outputs of the generic blocks. Simulators does not check compatibility of signals, hence there can be set, for example, electric current as voltage input for consequent blocks. For this purpose, so-called signal ontology is defined as well, being in fact a taxonomy of occurring signals.

For the integration and semi-automatic design methods, the relationship of the three ontologies plays an important role. Since each real plant device can be represented by more than one generic simulation block, the crucial issue is to represent explicitly the mapping between real devices and simulation equivalents (RI-3). The mapping is realized by the object property "simulates", that can be seen in Fig. 2, where a device "Resistor" is approximated by two generic simulation blocks: "Sim Resistor 1" and "Sim Resistor 2". We assume that each device is simulated by one or more simulation blocks. A more general situation, accepting device and simulation block mapping being $n{:}m$, would require a decomposition of the mapping ontology property. As there must be stored port numbers of the devices, it would be useful to define the predicate "is_connected_to" having four parameters, but it is possible neither in description logics nor in OWL DL, as the arity is two by definition. Trying to avoid technical and implementation details, we present a simplified situation where only one port is used.

The plant, simulation and signal ontologies are depicted in Fig. 2, where these ontologies with their mappings are shown in a compact form, briefly called "automation ontology". The rounded blocks represent ontology individuals, the rectangles are ontology classes. The upper part of the figure, filled in blue color, is common for all feasible automation systems, the other levels are domain-specific. On the left side of the figure, there is depicted the plant ontology, classifying devices into "Actuator" that comprises devices being active from a control point of view, "Passive Element" that cannot be controlled, "Disturbance" that are uncontrollable conditions and phenomena, "Measure point" being in domain of property "measures", and "Controller" performing control algorithms. The further level of the plant ontology depends on a type of the industrial plant, in Fig. 2, a simplified electrical circuit is used and its three devices are depicted. Note that real plant classes declare properties to express the interconnection of the devices, denoted in the figure by the label located on the left side of the class rectangles and properties comprising parameters of the devices, denoted by the

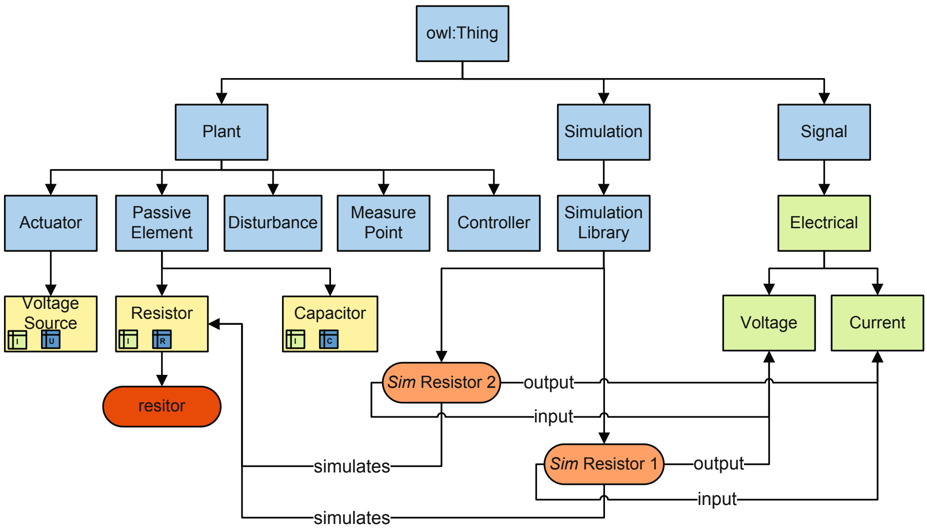


Fig. 2. Automation ontology

middle tags. Interconnections in the real plant domain are realized via ontology property called "power_bond", emphasizing that the real interconnection transfers energy. To reduce the size of the figure, only one real plant device is shown: a "resistor". The resistors are simulated by two generic simulation blocks, denoted by ontology property "simulates", that differ in their input and output signals. In a real case, the structure of both plant devices and generic simulation blocks is much more wider, as well as the signal taxonomy that involves only two electrical signals in this exemplary case.

5 Automatic Ontology-Based Selection of Blocks

Formalization of knowledge enables to use automatic approaches supporting many engineering tasks efficiently. In this chapter, we propose an ontology-based approach to assemble the simulation model, addressing the research issue RI-4. The entry for the algorithm are the plant ontology, comprising the real plant structure, and the simulation ontology, storing features of available generic simulation blocks. The goal of the automatical semantic engine is to find for each device (i.e. individual of the plant ontology) an equivalent generic simulation block with respect on compatible interfaces of blocks interconnected in series. The engine assembles the simulation model structure automatically, but block parameters are entered manually, hence the engine is referred as semi-automated.

To find appropriate simulation equivalents for each real plant device, there must be selected such blocks that simulate the particular devices. By this selection, we obtain a set of candidates for each plant device. The next step is to select the appropriate blocks out of the candidates sets, which is driven by a rule adopting the signal-oriented point of view: *For each signal, an output interface of its producer must be equal to the input interface of its consumer.* Although this rule seems simple at first, it provides powerful support for the simulation model design. The ideas were expressed in a SPARQL query, being depicted in Fig. 3, and the results for the exemplary project are shown in Fig. 6.

```
PREFIX ontology: <c:/Implementation#>
SELECT ?plant_device ?generic_block
WHERE {
 ?plant_device           ontology:is_connected_to ?plant_producer.
 ?plant_device           rdf:type                 ?plant_class.
 ?generic_block          ontology:simulates       ?plant_class.
 ?plant_producer         rdf:type                 ?plant_producer_class.
 ?producer_generic_block ontology:simulates       ?plant_producer_class.
 OPTIONAL {
 ?generic_block          ontology:input           ?interface1.
 ?producer_generic_block ontology:output          ?interface2.
 FILTER (?interface1 != ?interface2)
 } FILTER (!bound(?interface1))
}
```

Fig. 3. The SPARQL query to select appropriate generic simulation blocks from a universal simulation library

The SPARQL query depicted in Fig. 3 is motivated by the following steps:

1. Find all ontology individuals that represent the real plant devices. These individuals are recognized by a relation "is_connected_to".
2. For each real plant device, find its simulation equivalents, i.e. generic simulation blocks that "simulates" it.
3. For each device, find a producer of the signal being consumed by the device.
4. For each simulation block, find its input interface and an output interface of its input signal producer.
5. Select such simulation blocks, for which the interfaces are equal.

The current version requires the equality of interfaces, but we plan to transform the convertible variables and scales automatically in future work. Results of the SPARQL query depicted in Fig. 3 can be classified as follows:

1. *For each plant device exists exactly one simulation equivalent.*
   This solution is the most desirable; there exists exactly one simulation model of the industrial plant that is feasible and can be generated automatically.
2. *More than one solutions exist.*
   This class of solutions enables to create several simulation models for the industrial plants. Although all of them can be generated automatically, the engineer has to decide which of them is the most suitable for the task.
3. *A solution satisfying all conditions does not exist.*
   This result proves that the universal simulation library has to be modified.

6 Use-Case: Simplified Electrical Circuit

To demonstrate the presented approach, a simplified electrical circuit was selected as a use-case. It consists of three devices: an ideal source of voltage, an ideal resistor, and an ideal capacitor. Although the methods are oriented on large-scale industrial systems, such a simple case enables humans to check the results and furthermore, it can be a sub-part of some complex cyber-physical system, i.e. a system comprising segments of diverse engineering disciplines. Fig. 4 depicts the schema of the circuit on the left side and the adequate plant ontology individuals on the right side.

The output value of the ideal source of voltage is the constant voltage independent on the output current. The output of the ideal capacitor is voltage and the input is electric current. Although a capacitor could be also modeled inversely, having voltage as its input and electric current as the output, this model would not be causal, because the mathematical description would involve a derivation, hence this approximation is not considered. The simulation model

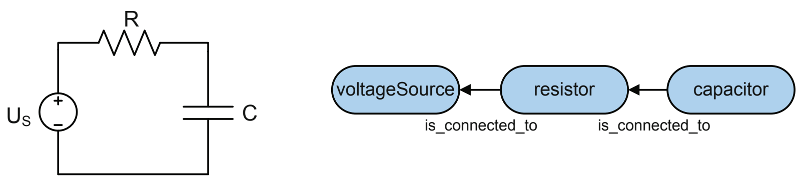


Fig. 4. Use-case electrical circuit: Schema and its representation with plant ontology individuals

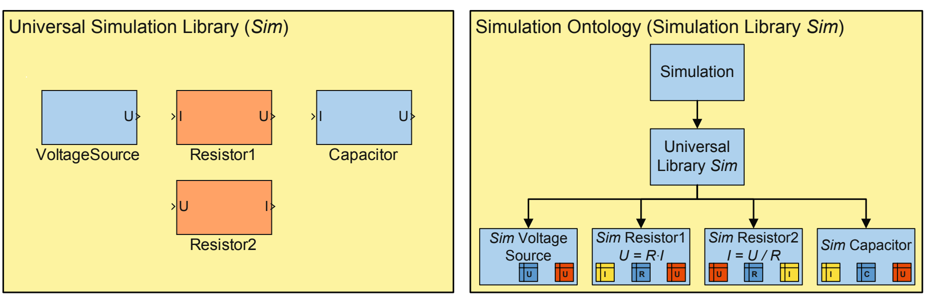


Fig. 5. Simplified simulation library with the dominant interfaces and representation of its semantics with simulation ontology

plant_device	generic_block
`voltageSource`	`sim_voltage_source`
`resistor`	`sim_resistor2`
`capacitor`	`sim_capacitor`

Fig. 6. Result of the SPARQL query. For each plant device the appropriate generic simulation block is selected.

of a resistor is much more interesting, because this element has no restriction, how to be simulated. The universal simulation library involves two resistors. *Resistor 1* has electric current as its input, voltage as output and `R` as a tunable parameter. Its behavior is given by $u = Ri$; whereas *Resistor 2* has a transfer function $i = u/R$, i.e., voltage is the input and electric current is the output.

The semantic information about available simulation blocks, their parameters, and interfaces are comprised in the simulation ontology, as depicted in Fig. 5. In such an extremely simplified example, it is very fast and easy to recognize that *Resistor 2* must be used for the simulation of this electric circuit, but in real large-scale systems, such decisions are time consuming and errors can occur. The SPARQL query was used for this task and its results are depicted in Fig. 6.

7 Conclusion and Future Work

The paper deals with using ontologies and ontology querying to improve the design phase of simulations as one of the important tools for optimization of automation systems and industrial plants operations. The presented approach is based on Semantic Web technologies providing efficient formalization of knowledge related to heterogeneous data-models and enabling to perform ontology-based methods, such as ontology querying or reasoning. Knowledge related to a structure of the real plant is represented in the plant ontology, knowledge about available simulation blocks is stored in the simulation ontology, and the taxonomy of interacting signals is formalized in the signal ontology. For various engineering tasks, a mapping between these ontologies plays an important role. The plant and simulation ontologies are mapped via the object property "simulates" that defines which real plant device is simulated by particular simulation blocks. The mapping between simulation and signal ontologies is realized by object properties "input" and "output" that define the interfaces of the blocks.

The proposed method is based on the selection of appropriate simulation blocks via the SPARQL query given in the paper. The query identifies a set of candidate generic simulation blocks that simulate each plant device. Consequently, the appropriate simulation block is selected out of the candidate set by matching the type of the input signal with the signal type of its producer. The explained method is demonstrated using the practical example dealing with a simplified electrical circuit. In this example, two generic simulation blocks for a resistor are available in the universal simulation library and it is demonstrated that the query finds the correct solution.

The ontology-based representation of the knowledge about industrial plants and the overall automation project provides flexible data representation, enables to use sound methods, such as ontology querying or reasoning, and can notably support both system design and maintenance. The early project stages dealing with the ontology structure design and definition could seem quite complex and time consuming, but the reusability and semantic representation enhance standardization and lower time and costs for maintenance or redesign. The main limitation of the proposed method is the current implementation for single-input and single-output blocks that are connected in series.

Future work will focus on the practical application of this approach in order to estimate its computational complexity for real large-scale systems. We plan further generalization to support multi-input and multi-output interfaces as well as the parallel interconnections of the blocks. We also plan to support identification of missing blocks and generation of their interfaces automatically. As it has been explained in the text, the automatic conversion between similar interfaces remains another future work issue.

Acknowledgments. The authors would like to thank their partners from the Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems for the discussion and feedbacks. This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria.

References

1. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering with examples from the areas of Knowledge Management. In: e-Commerce and the Semantic Web. Springer, London (2004) (second printing)
2. Obitko, M., Mařík, V.: Ontologies for Multi-agent Systems in Manufacturing Domain. In: Hameurlain, A., Cicchetti, R., Traunmüller, R. (eds.) DEXA 2002. LNCS, vol. 2453, pp. 597–602. Springer, Heidelberg (2002)
3. Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2) (1993)
4. Silver, G., Hassan, O.H., Miller, J.: From domain ontologies to modeling ontologies to executable simulation models. In: Proc. of the 2007 Winter Simulation Conference, pp. 1108–1117 (2007)
5. Lastra, J.L.M., Delamer, I.: Ontologies for Production Automation. In: Dillon, T.S., Chang, E., Chung, S., Sycara, K. (eds.) Advances in Web Semantics I. LNCS, vol. 4891, pp. 276–289. Springer, Heidelberg (2008)
6. Moser, T., Mordinyi, R., Sunindyo, W., Biffl, S.: Semantic Service Matchmaking in the ATM Domain Considering Infrastructure Capability Constraints. In: Canadian Semantic Web: Technologies and Applications. Springer, Heidelberg (2010)
7. Miller, J.A., Baramidze, G.: Simulation and the semantic Web. In: Proc. of the 2005 Winter Simulation Conference (2005)
8. Moser, T., Biffl, S.: Semantic tool interoperability for engineering manufacturing systems. In: Proc. of the IEEE Conference on Emerging Technologies and Factory Automation, ETFA (2010)
9. Cheng, S.-Y., Shen, B.-C., Peng, M.-J., Tian, Z.-F., Zhao, Q., Xue, R.-J., Gong, C.: Research on coordinated control in nuclear power plant. In: International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3622–3627 (2009)

Fact-Based Service Modeling in a Service Oriented Architecture

Peter Bollen

Maastricht University
Faculty of Economics and Business Administration
Department of Organization & Strategy
P.O. Box 616, 6200 MD Maastricht, The Netherlands
p.bollen@maastrichtuniversity.nl

Abstract. Service-oriented computing (SOC) allows organizations to tailor their business processes, in such a way that efficiency and effectiveness goals will be achieved by outsourcing (parts of) business processes to external (web-based) service-providers. In order to find the computing service-providers that provide the organizations with the biggest benefits, it is paramount that the service-requesting organization (SRO) has a precise description of the service it wants to have delivered by the service delivering organization (SDO). In this paper we will illustrate how enterprises that play the SDO and SRO roles can be conceptually integrated by creating conceptual models that share the definitions of the business processes within the service oriented architecture (SOA) framework.

Keywords: Service-oriented architecture, SOA, fact-orientation, ORM.

1 Introduction

In the service-oriented architecture (SOA) paradigm, a service requesting organization (SRO) basically outsources one or more organizational activities or even complete business processes to one or more service delivering organizations (SDOs). The way this is done currently, is that the SRO 'outsources' a given computing service to a 'third-party' SDO for a relative long period of time (3 months, a year). The selection and contracting activities are performed by organizational actors, i.e. managers responsible for the business processes in which the service(s) is (are) contained. Most of the current SDO's provide 'internet substitutes' [1] for functions that used to be performed by an (integrated) SRO's enterprise system, implying that the SRO's that use these process services are shielded from the intrinsic complexities of these 'substituted' functionalities [2, 3].

Current approaches for web services, have limitations on a semantic and ontological level (among others) [4]. The problem with current approaches is that they cannot handle the semantic and ontological complexities caused by flexible participants having flexible cooperation processes. Enterprise integration between participants (i.e. broker, SROs and SDOs) can only be achieved if the conceptual schema of the content, e.g. it's ontology can be expressed formally and explicitly [5].

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 158–169, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The application of the service-oriented paradigm that will lead to the most benefits for the SRO will be embedded in a semantic-web environment in which the 'outsourcing' decision in principle, can be made in real-time every time a service is requested [6]. This real-time level of decision making implies that the service-processes that are requested should be defined in such a way that the negotiation, contracting and execution of the service can take place in 'run-time' without 'design time' human intervention. In fact-based terminology we can say that a process is a fact-generating activity [7].

In this paper we will apply the fact-based conceptual modeling language (e.g. as documented in several variations in [8-12]), that will enable businesses to define their platform independent models for their service-oriented requirements and that will allow for an 'automatic' transformation from the platform independent model (PIM) to the platform specific model (PSM) levels [13, 14].

2 Related Work

2.1 Related Work on Conceptual Modeling Approaches

In compliance with the emerging OMG standard on business rules: SBVR [15], we will consider a family of conceptual modeling approaches that traditionally have been used for the specification of relational databases [11] but have evolved into business rule modeling languages [16-20]. We will refer to this family of approaches as fact-based modeling approaches. The latest extensions of the fact-based modeling approaches cater for the declarative modeling of business processes [21-23] and business events [22, 24]. In an earlier article the application of fact-orientation for applying web ontologies has been discussed [25].

The (extended) fact-based approach structures verbalizable knowledge into the following elements [26]:

1. Knowledge domain sentences
2. Concept definitions and naming conventions for concepts used in domain sentences
3. Fact types
4. Fact type readings for the fact types
5. Population state (transition) constraints for the knowledge domain
6. Derivation rules that specify *how* specific domain sentences can be derived from other domain sentences.
7. Exchange rules that specify *what* fact instances can be inserted, updated or deleted.
8. Event rules that specify *when* a fact is derived from other facts or when (a) fact (s) must be inserted, updated or deleted.

The combined Knowledge Reference Model (KRM) consisting of elements 1 through 8 of the above captures a complete description of a domain- or application's conceptual schema, including the domain- or application ontology (elements 2, 3 and 4).

A legend of the ORM-(I) notation used in this article is provided in the following. The 'role-based' ORM notation makes it easy to define static constraints on the data structure and it enables the modeler to populate ORM schemas with example sentence instances for constraint validation purposes. In ORM (and other fact-based approaches) the fact construct is used for encoding all semantic connections between entities. Figure 1 summarizes the symbols in the ORM modeling language that we have used in this paper.

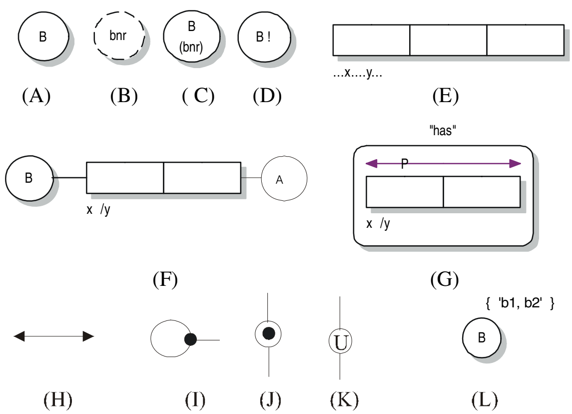


Fig. 1. Main symbols in Object-Role Modeling (ORM)

Atomic *entities* (figure 1A) or *data values* (figure 1B) are expressed in ORM as simple (hyphenated) circles. Instances of an entity type furthermore can exist independently (e.g. they are not enforced to participate in any relationship), which is shown by adding an exclamation point after the entity type's name (figure 1D). *Simple* reference schemes in ORM are abbreviated by putting the *value type* or *label type* in parenthesis beneath the name of the entity type (figure 1C). Semantic connections between entities are depicted as combinations of boxes (figure 1E) and are called *facts* or *fact types* in ORM. Each box represents a role and must be connected to either an *entity type*, a *value type* or *a nested object type* (see figure 1F). A fact type can consist of one or more roles. The number of roles in a fact type is called the fact type arity. The semantics of the fact type are put in the *fact predicate* (this is the text string *…x…y…* in figure 1E). A *nested object type* (see figure 1G) is a non-atomic entity type that is connected to a fact type that specifies what the constituting entity types and/or values types are for the nested object type. Figures 1H through 1L illustrate the diagramming conventions for a number of *static population constraint(s) (types)* in ORM.

2.2 Related Work on Process Modeling

The research on the *process-oriented perspective* in the information systems research community was prominent in the late seventies and has resulted in numerous process-oriented information systems development methodologies like SADT [27] and SA/SD [28] that are still in use. Although these development methodologies were process-oriented they also contained modeling constructs for the data-oriented perspective. In the eighties a number of system development methodologies were proposed that covered both the data-oriented, process-oriented and behaviour-oriented perspectives [29-31]. In the nineties a research school on 'workflow management' emerged (see for a good literature review [32]). Around that time the business process reengineering 'paradigm' [33, 34] in combination with the increasing popularity of Enterprise Resource Planning packages (e.g. SAP, see [35]), lead to the development of domain-oriented analysis methods [36] of which the ARIS-based Business Process Modeling [37, 38] and BML [39] are good examples. A recent standard for expressing business processes is the business process modeling notation specification BPMN [40].

2.3 Related Work on Service-Oriented Architecture

In [41] a SOA (service oriented architecture) is provided. The basic elements from this service-oriented approach to distributed software design is given in figure 2. In the SOA from figure 2, service delivering organizations (SDOs) or service providers use the registry service (or broker [5] or service repository [47]) to publish their identity and a description of services that they provide. When a service requesting organization (SRO), service requestor [47] or service client, needs a service, it queries the lookup service (service discovery [48]) which will initiate the communication between SRO and SDO to establish a commitment regarding the service delivery [48].

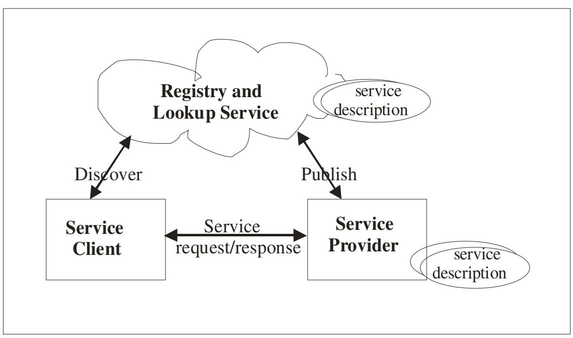


Fig. 2. SOA architecture as given in [41] and [14]

2.4 Related Work on the Modeling of Service-Oriented Enterprise Architectures

The specific paradigm that is center of the discussion in this article is the structuring of businesses according to services that are provided or needed [42]. Research that integrates conceptual (meta)modeling and the SOA paradigm mostly relates to the meta level, especially the meta process model, e.g. as in Piprani et al. [43] and the ontology part of the meta model, e.g. Terlouw [44]. Other researchers investigate, how business rule management can be extended to the service-oriented paradigm [45]. We will take the SOA architecture from figure 2 as a basis for our further application of conceptual modeling on the SOA domain. We will thereby, explicitly distinguish three universes of discourse (UoD's): the client (SRO), the service provider (SDO) and the broker (or registry and look-up service).

3 Fact-Based Conceptual Modeling of the SRO

We will extend the current modeling capabilities of the fact-based approach with modeling constructs for the modeling of business services in the context of the service-oriented paradigm by extending the concepts definitions and derivation/exchange rule modeling constructs to cater for 'business services' that can be provided by either the SRO itself or by one or more (external) SDO(s).

In order to use the semantic web for selecting and contracting SDO's for any business function that needs to be outsourced, business organizations need conceptual modeling tools that define these functions or (parts of) business processes. The commonly used process modeling approaches lack the capabilities to be used for this purpose [23]. In this section we will extend the fact-based conceptual modeling approach to cater for the definition of business functions (or parts of business processes) during design time in such a way that in a semantic web environment in which SRO and SDO's can interchange their domain ontologies and thereby in run-time can decide which of the relevant SDO's will be partner to deliver the requested service for a given business transaction.

We will present the elements from the fact-based knowledge reference model (KRM) and see how they can be applied in the situation in which SDOs are involved in (interorganizational) business processes. We will use as a running example for the UoD of the SRO, the (fictitious) ABC company, and focus on the carrier selection process for customer shipments.

3.1 The SRO UoD: The ABC Company's Carrier Selection Business Process

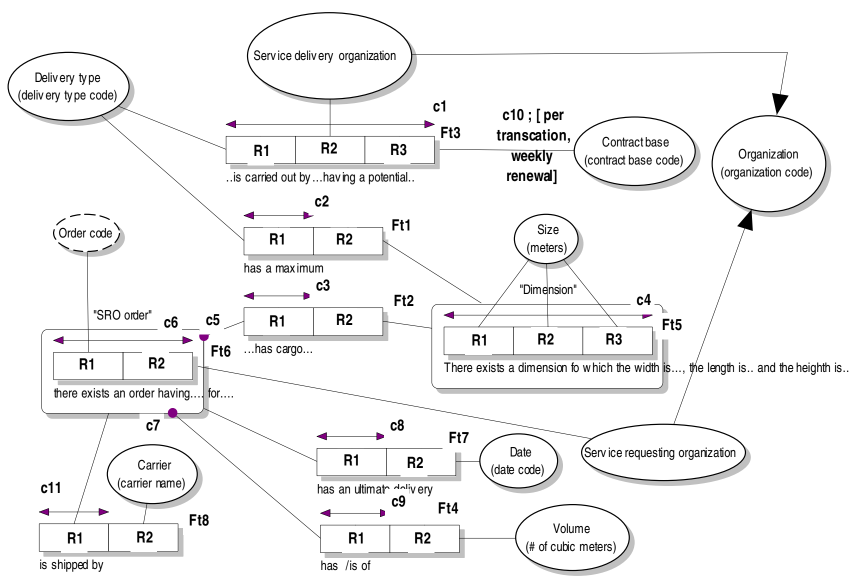
ABC is a business that operates a number of 'brick-and-mortar' stores. Although the company does have an internet retail-website, it sometimes receives order request for deliveries via mail, e-mail or fax, outside the sales region it serves and in some cases even outside the country it operates in, and sometimes it receives 'overseas' order requests. Especially for the latter order category, ABC can make an additional profit by shipping the order using the cheapest carrier at any given point in time. The shipping fee, they charge to their customer is a constant fee. The customer has the choice between standard shipping and express shipping. The ABC company, has a logistics department in which 1 person is responsible for the shipment of continental and overseas orders. Since this person, has also other logistics responsibilities, he/she can not afford to spend too much time trying to search for the best transportation deals. It might be beneficial for ABC, to 'outsource' the carrier selection process to a third-party, in this case a service delivery organization (SDO).

We will start the presentation of our fact-based model by providing a list of structured concept definitions. This list of structured concept definitions, should facilitate the comprehension of knowledge domain sentences and comprise the business domain ontology [46].

Table 1. Excerpt of List of concept definitions for SRO

Concept	Definition
Organization	A business entity that delivers services and/or goods to customers and/or other business entities.
Organization code	A name from the *organization code* name class that can be used to identify an [organization] among the set of [organization]s.
Dimension	Size of [cargo] as length* width * height
Delivery type	A generally agreed upon type of delivery by a [service requesting organization] and a service registry organization or broker that is characterized by a maximum [dimension]
Contract base	Type of commitment between a [service delivery organization] and a [SRO]
'Per transaction' contract base	A specific value for a [contract base code] that means that a contract between a [SDO] and a [SRO] change per transaction on the discretion of a [SRO].

In case a standard between the SDO's and SRO's and service broker has been implemented, in which it is agreed upon that*: for any (predefined) delivery type at most one maximum dimension can exist*, we can show this as a uniqueness constraint of fact type Ft1 that covers the role R1. A further formalization of the allowed communication within the SRO's UoD's is the convention that *a given order by a given SRO must have exactly one dimension*. The latter business rule is encoded in the



Define Order has Volume (cubic meters)**as** Order has cargo Dimension **and**
There exist a dimension for which the width is $Size_1$ and the length is $Size_2$ and the heighth is $Size_3$ **and**
Volume= $Size_1$ * $Size_2$ * $Size_3$

Fig. 3. Complete conceptual schema for SRO (in combination with table 1)

ORM fact type model in figure 3 as a uniqueness constraint spanning role R1 of fact type Ft2 in combination with a mandatory role on the nested entity type SRO-order. Finally, in role R3 of fact type Ft3 we can define a value constraint in which the allowed and enumerable set of values can be listed.

3.2 Derivation Rules

In addition to the business rules that can be expressed as population state constraints, we can add business rules that can derive 'new' fact instances from 'old' fact instances. An example of such a derivation rule can be applied for fact type *Ft4*. We assume that *a volume is the multiplication of the three dimensions* figures that are modeled in fact type *Ft2/Ft5*. This derivation rule can be modeled as a derivation rule in figure 3 in which formula: Ft4.R2 = Ft5.r1*Ft5.r2*Ft5.r3 is contained.

We note that in a service oriented architecture, derivation rules play an important role because SRO's 'outsource' the execution and management of these rules to SDO's. It's therefore, paramount to incorporate the definition of these derivation- and exchange processes into the list of concept definitions (see table 2). We can see that the process *calculate volume* is implemented within the sphere of influence of the organization itself. The process is made explicit in the form of derivation rule: *Define order has Volume (cubic meters),* that is listed at the bottom of figure 3. The process determine carrier for order, however is outsourced to some SDO.

Table 2. Excerpt of List of concept definitions for SRO

Process: Calculate Volume	A process that has a a result: a rough indicator of the cubic [volume] of a package which is determined by multiplying its width, heighth and length. **<Create(s) instance(s) of Ft4>**
Process: Add order	A transaction in which the [order] and the [dimension] and [delivery date] of the [order] are added to the information system. **<Create(s) instance(s) of Ft2 and Ft7>**
Process: Determine carrier for order	This process leads to the selection of a specific [SDO] for the shipment of an [order] under the best possible conditions for [delivery time] and [shipment price] **<Create(s) instance(s) of Ft8>**

4 Fact-Based Conceptual Modeling of the SDO

In this section we will look at the Universe of Discourse of a web-service that provides carrier selection services for SRO's. One of the main processes within this UoD's is the up-to-date acquisition of carrier data regarding latest offers, in terms of shipment conditions, and prices for each delivery type and possibly delivery (sub)-types depending upon each individual carrier. This web-service organization has as objective to match SRO's with carriers normally for a small fee per transaction. We will see that ontological commitments need to be established between SRO's and SDO's on a 'design'-time level. This means that key concepts for web-based service transactions will be harmonized (as can be checked for example in the list of concept definitions in tables 1 and 3, for the concept *delivery type* and *carrier*).

Table 3. Excerpt of List of concept definitions for SDO

Concept	**Definition**
Carrier	A third party logistics organization that ships packages for an [order] from a [SRO] to a client of the [SRO]
Carrier delivery type	A [local delivery type] that is offered by a [carrier]
Process: Classify service offering	A process that has a result a classification for a [local delivery type] offered by a [carrier] in terms of an instance [delivery type] that has been defined by a [SRO] and [SDO].**<Create(s) instance(s) of Ft108>**
Process: Add service offering delivery length	A process that has a result that a maximum delivery length for a [carrier delivery type] is entered into the information base. **<Create(s) instance(s) of Ft106>**
Process: Add service offering standard price	A process that has a result that a [standard price] for a [carrier delivery type] is entered into the information base **<Create(s) instance(s) of Ft101>**
Process: Add service offering promotional price	A process that has a result that a [promotional price] during one or more [weeks] for a [carrier delivery type] is entered into the information base **<Create(s) instance(s) of Ft102 and Ft107>**

On the other hand, promotional concepts and other rating schemes can be introduced on the fly, at any time by a carrier. For many of these promotional campaigns and or new tariff schemes, it will not be feasible to establish ontology harmonization between the SDO and these carriers at all times. To cater for this, we need modeling constructs that allow us to deal with the runtime changes in domain concepts as used by SDO's in their carrier selection processes on behalf of their SRO customers. We will show now in our example list of definitions and conceptual schema for the UoD of the SDO can be modeled for these short-term runtime definitions of domain concepts. We note that a 'snaphot' of delivery types for every carrier that that is considered by a carrier-selection SDO will be modeled as a populations of fact type Ft1 in the conceptual schema of the carrier selection SDO in figure 4. We now see that the carrier selection broker service not only provides the best deal for a service requesting organization, but also performs the role of 'ontological harmonizer' between the SRO's and the carriers by introducing and defining the concepts of local delivery type and carrier delivery type.

In table 2 we have provided the extended list of concept definitions for this example UoD of a service delivery organization in which the definitions of the fact generating processes are incorporated. If we inspect the conceptual schema for our fictitious example SDO in figure 4, we can say that one of the 'core business processes' for the SDO, is the establishing, ontological harmonization, in 'run-time' this means that the SDO will populate fact type Ft108 in figure 4, by continuously scanning for recent service offerings provided by existing and new carriers. This business process mainly scans and interprets these service offerings, and as a result will 'label' these offerings and subsequently classify them, in the terminology, that was established between the SRO's and SDO's, via a broker or registry service. In the running example of this article, we have limited ourselves to only depict a few relevant fact types that will be used in practice. In a real-life conceptual schema 100's of fact types might actually be used in the communication, between SDO, SRO and registry service.

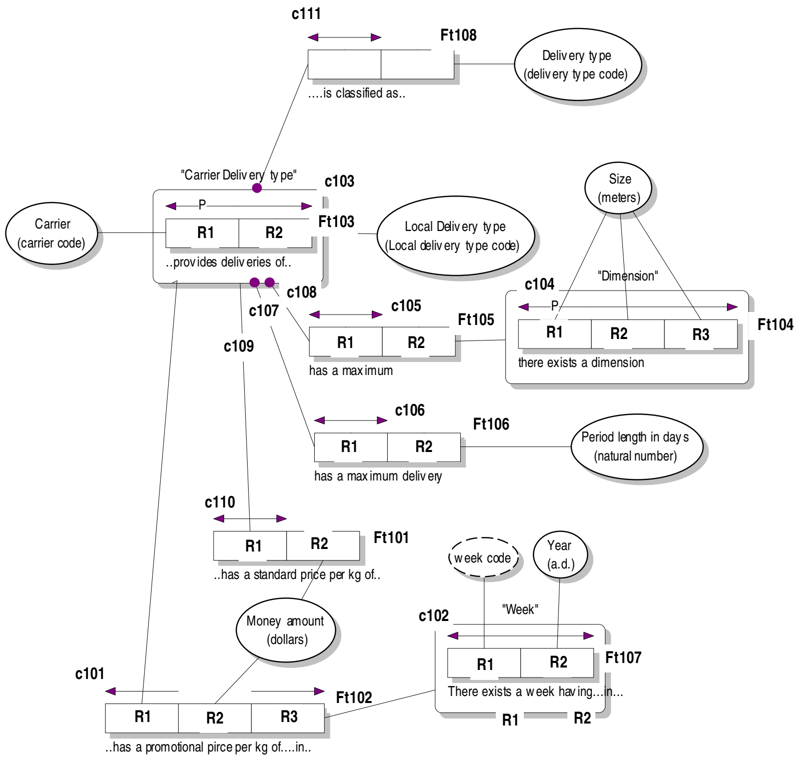


Fig. 4. Conceptual schema for SDO (in combination with table 2)

5 Conclusion

In line with semantic web developments, the conceptual schema needs a communication part that contains 'definition' instances to be shared with the potential agents in order for them to be able to communicate effectively and efficiently with a ('web-based') business application. Adding these semantic definitions of business processes is a requirement for achieving enterprise integration. This will significantly increase the quality and ease-of-use of such a (web-based) application, since it has established a semantic bridge with the potential external users, allowing them to communicate in a direct way with the business application, by preventing semantic ambiguities from occurring in the first place. Another advantage of applying fact-orientation for capturing an application or a (relatively complex) domain's ontology is in its flexibility to use it even to model communication between agents in which (explicit) ontological harmonization at a type or schema level is not possible or desirable. By adding 'run-time' concepts as populations of (typed) concepts for which an ontological harmonization already has been established.

Another advantage of using (extended) fact-based modeling languages is that a business organization is not forced to remodel the application or domain ontology every time a new ‘implementation’ standard has been defined. Business organizations can capitalize on the fact-based conceptual modeling investment, for the foreseeable future by applying the appropriate mappings between a fact-based application ontology and the implementation standard of the time.

References

1. Siau, K., Tian, Y.: Supply chains integration: architecture and enabling technologies. Journal of Computer Information Systems 45, 67–72 (2004)
2. Estrem, A.: An evaluation framework for deploying web services in the next generation manufacturing enterprises. Robotics and Computer Integrated Manufacturing 19, 509–519 (2003)
3. Baina, K., Benali, K., Godart, C.: Discobole: a service architecture for interconnecting workflow processes. Computers in Industry 57 (2006)
4. Shen, W., et al.: An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. Robotics and Computer Integrated Manufacturing 23, 315–325 (2007)
5. Yue, P., et al.: Semantics-based automatic composition of geospatial web service chains. Computers & Geosciences 33, 649–665 (2007)
6. Menascé, D., Ruan, H., Gomaa, H.: QoS management in service-oriented architectures. Performance Evaluation 64, 646–663 (2007)
7. Bollen, P.: Conceptual process configurations in enterprise knowledge management systems. In: Applied Computing 2006. ACM, Dijon (2006)
8. Verheijen, G., van Bekkum, J.: NIAM: An Information Analysis method. In: IFIP TC-8 CRIS-I Conference. North-Holland, Amsterdam (1982)
9. Bakema, G.P., Zwart, J.P., van der Lek, H.: Fully communication oriented NIAM. In: Nijssen, G., Sharp, J. (eds.) NIAM-ISDM 1994 Conference, Albuquerque NM, p. L1–35 (1994)
10. Nijssen, G., Halpin, T.: Conceptual schema and relational database design: A fact based approach. Prentice-Hall, Englewood Cliffs (1989)
11. Halpin, T.: Information Modeling and Relational Databases; from conceptual analysis to logical design. Morgan Kaufmann, San Francisco (2001)
12. Lemmens, I.M.C., Nijssen, M., Nijssen, S.: A NIAM2007 Conceptual Analysis of the ISO and OMG MOF Four Layer Metadata Architectures. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 613–623. Springer, Heidelberg (2007)
13. OMG, MDA guide version 1.0.1. In: Miller, J., Mukerji, J. (eds.) OMG (2003)
14. Jardim-Goncalves, R., Grilo, A., Steiger-Garcao, A.: Challenging the interoperability between computers in industry with MDA and SOA. Computers in Industry 57, 679–689 (2006)
15. OMG, Semantics of Business Vocabulary and Business Rules (SBVR), first interim specification, p. 392 (2006)
16. Nijssen, G.: Semantics for Business: a process to specify the most important business rule in SVBR. Business Rules Journal 8(12) (2007)
17. Nijssen, G.: SBVR: Semantics for Business. Business Rules Journal 8(10) (2007)
18. Nijssen, G.: Kenniskunde 1A. PNA Publishing Heerlen (2001)
19. Halpin, T.: Business Rule Modality. In: Proc. EMMSAD 2006: 11th Int. IFIP WG8.1 Workshop on Exploring Modeling Methods in Systems Analysis and Design (2006)

20. Halpin, T.: A Fact-oriented Approach to Business Rules. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) ER 2000. LNCS, vol. 1920, pp. 582–583. Springer, Heidelberg (2000)
21. Bollen, P.: Fact-Oriented Business Service Modeling. In: 12th international workshop on Exploring Modeling Methods in Systems Analysis and Design (EMSSAD 2007), Trontheim, Norway (2007)
22. Bollen, P.: Fact-oriented Business Rule Modeling in the Event Perspective. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495. Springer, Heidelberg (2007)
23. Morgan, T.: Business Process Modeling and ORM. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 581–590. Springer, Heidelberg (2007)
24. Bollen, P.: Fact-oriented Modeling in the Data-, Process- and Event Perspectives. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 591–602. Springer, Heidelberg (2007)
25. Bollen, P.: Natural Language Modeling for Business Application Semantics. Journal of Information Science and Technology 2(3), 18–48 (2005)
26. Nijssen, G., Bijlsma, R.: A conceptual structure of knowledge as a basis for instructional designs. In: The 6th IEEE International Conference on Advanced Learning Technologies, ICALT 2006, Kerkrade, The Netherlands (2006)
27. Connor, M.: Structured analysis and design technique. Waltham (1980)
28. Yourdon, E., Constantine, L.: Structured design: Fundamentals of a discipline. Yourdon-Press (1979)
29. Flynn, D., Warhurst, R.: An empirical study of the validation process within requirements determination. Information Systems Journal 4, 185–212 (1994)
30. Olive, A.: Dades- a methodology for specification and design of information systems design and management. In: Information Systems Design Methodologies- a Comparative Review, North-Holland (1982)
31. Rolland, C., Richard, C.: The REMORA methodology for Information Systems Design and Management. In: IFIP WG8.1 Working Conference on Information Systems Design Methodologies: a Comparative Review, North-Holland (1982)
32. Salimifard, K., Wright, M.: Theory and Methodology: Petri net-based modelling of workflow systems: an overview. European Journal of Oper. Res. 134, 664–676 (2001)
33. Davenport, T., Short, J.: The new industrial engineering: information technology and business process redesign. Sloan Management Review, 11–27 (Summer 1990)
34. Hammer, M.: Reengineering work: don’t automate, obliterate. Harvard Business Review, 104–112 (July-August 1990)
35. Curran, T., Ladd, A.: SAP R/3 business blueprint. Prentice-Hall (2000)
36. Nissen, H., Jarke, M.: Repository for multi-perspective requirements engineering. Information Systems 24, 131–158 (1999)
37. Scheer, A.: Business Process Engineering: reference models for industrial enterprises. Springer, Berlin (1998)
38. Scheer, A.: ARIS_ Business process modeling, 2nd edn. Springer, Berlin (1999)
39. Johannesson, P., Perjons, E.: Design principles for process modeling in enterprise application integration. Information Systems 26, 165–184 (2001)
40. OMG, Business process modelling notation (BPMN) OMG available specification v 1.1., OMG (2008)
41. McIntosh, R.: Open-source tools for distributed device control within a service-oriented architecture. Journal of the Association for Laboratory Automation (9), 404–410 (2004)

42. Engels, G., et al.: A method for engineering a true service-oriented architecture. In: 10th International Conference on Enterprise Information Systems, Barcelona, Spain (2008)
43. Piprani, B., Wang, C., He, K.: A Metamodel for Enabling a Service Oriented Architecture. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 668–677. Springer, Heidelberg (2008)
44. Terlouw, L.: Ontology-based change management of composite services. In: Workshop on Pervasive Systems, Montpellier, France (2006)
45. Weigand, H., van den Heuvel, W.-J., Hiel, M.: Rule-based service composition and service-orienetd business rule management. In: International Workshop on Regulations Modelling and Deployment (ReMoD 2008), Aachen, Germany (2008)
46. Bollen, P.: Extending the ORM conceptual schema design procedure with the capturing of the domain ontology. In: EMMSAD 2007, Tapir. Academic Press, Trondheim (2007)

A Lightweight Data Integration Architecture Based on Semantic Annotated RSS Feeds

Amin Anjomshoaa, A Min Tjoa, and Homa Rezaiee

Vienna University of Technology,
Institute of Software Technology and Interactive Systems,
Favoritenstr. 9-11, 1040 Vienna, Austria
{anjomshoaa,amin}@ifs.tuwien.ac.at, rezaieehoma@gmail.com

Abstract. Web 2.0 has changed the technological landscape of the Internet computing world today by supporting efficient methods for information sharing and social collaboration. Web 2.0 is much more than adding a nice facade to old web applications rather it is a new way of thinking about software architecture of RIAs which opens new possibilities for finding more effective answers to old problems such as data integration. In this paper a methodology for integration of distributed information based on semantically-enabled RSS (Really Simple Syndication) feeds will be presented. The proposed method can be used in multiple use-cases where service providers/end-users need ad-hoc exchange of their service information/requirements in order to find an optimized solution. As proof of concept an emergency management use-case cases is presented which shows how semantic web concepts can be embedded in RSS feeds to address the specific requirements of accident and emergency scenarios.

Keywords: Semantic Web, Web 2.0, GeoRSS, Data Integration.

1 Introduction

Integration of distributed and heterogeneous data resources is one of the classical problems in information technology and its importance for effective information management has risen along with the data-explosion phenomenon. Today, there are a handful of information management tools, such as middleware, data convertors, and Mashups that are to some extent able to couple heterogeneous data sources together; however most of the data integration tasks require intensive human intervention. In other words the existing data integration methods address the integration of data structures but semantic of data integration and resolving the logical conflicts should be managed by human user.

Data integration is also widely used in business use-case where the business analysts have to deals with a highly interdisciplinary network of knowledge resources that are coming from different domains. Generally the analysis of scattered business information and deploying them in a different business solution is not conceivable without huge contribution of human users who take over the complex task of information integration. As a matter of fact, a big portion of IT processes deal with the information integration issues which are necessary to shape the information for

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 170–177, 2011.
© Springer-Verlag Berlin Heidelberg 2011

specific use-cases. Usually such information integration processes are created by IT experts and presented to the end-users as a service; however, this situation is being changed by emergence of Web 2.0 concept which has hit the mainstream, and continues its rapid evolution.

Successful Web 2.0 applications are characterized by crowd-sourcing and social networking where people create and share the content and services. Web 2.0 makes better use of the client machine's processing power and at the same time pushes the SOA paradigm to its limits. In this context, Internet will play the role of a global operating system that hosts the Web Services. In other words, the Web 2.0 fosters the global cloud computing idea where business services are presented on Internet and developers should select and weave them together to create new compound services [1]. Web 2.0 has also introduced new possibilities for a better human computer interaction and data sharing via machine-readable data feeds in Atom and RSS formats. By publishing the data in one of these formats, remote clients can read and integrate this data into their processes. As a result, this lightweight architecture can be used to implement a loosely coupled data integration for distributed resources which looks to be more appropriate for ad-hoc and dynamic data models. This data integration model can be especially helpful for modern business use-cases with highly dynamic and transient data models that are synced and updated based on distributed information sources. In this regard Web 2.0 paradigm has offered new possibilities for lightweight data integration.

The web feeds can be also enriched with further context information in order to support more efficient data processing. For instance an RSS data feed can also include geographical information. This specific form of RSS which is known as GeoRSS is an emerging standard for encoding location as part of web feeds [2].

In this paper we will present a lightweight data integration architecture based on GeoRSS feeds which are enriched with semantic annotations. As proof of concept an emergency management use-case for addressing accident requirements is selected where several service providers such as hospitals, police, and fire brigade needs to share their service descriptions in order to find the appropriate solution for handling an accident.

2 Related Work

Despite all advantages of SOA and Web Services for data integration purposes, they are complex technologies which need thorough understanding of these technologies and their underlying standards. Furthermore the SOA-based approaches for data integration is unable to react rapidly to the changes in the environment, as implementation of such systems are cost and time intensive and any changes in the environment may require some modifications in the system.

Web 2.0, on the other hand, has introduced new possibilities for efficient human computer interaction via rich applications such as Mashups that provide a user-driven micro-integration of web-accessible data [3]. Mashup envisions building effective and light-weight information processing solutions based on the exposed Web Services of organizations. Such Web Services may range from simple services such as RSS-based [4] and REST-based [5] services to complex BPEL services for more serious use-cases. The Web Feeds plays an important role in many Mashup platforms such as

IBM Mashup Center [6] where feeds are used to unlock information sources by creating consumable feeds from enterprise sources and a wide variety of departmental and desktop resources.

The RSS feeds as a content syndication method has already proven to be a powerful solution for content distribution and reaching a larger audience [7]. The RSS feeds are commonly used for person-to-person communication. In this pattern a publisher posts a blog item and then the subscribers are alerted. A growing number of people are also using RSS for process-to-person communication where users subscribe to a data resource which is dynamically updated. Subscribing to searches is the best example of this pattern [8]. A more advanced use of RSS was introduced by Amazon's OpenSearch that allows publishing of search results in a format suitable for syndication and aggregation [9]. In other words, the OpenSearch introduces a process-to-process pattern for using RSS feeds.

There is already some research works in emergency management that applies Semantic Web concepts to geographically distributed data sources and services in order to make them available through Semantic Web Services and facilitate semantic reasoning and inference [10]. However to the best of our knowledge there is no research toward studying the semantically-enabled RSS feeds and their applications and in our belief, RSS model like any other information management system can gain advantage from semantic technologies.

3 Emergency Management Use-Case

In response to major incident, different emergency services such as hospital, police, fire brigade, ambulance service, and local authorities must perform their designated roles according to their assigned responsibilities, as well as co-ordinate effectively with each other and with volunteers to ensure maximal efficiency in saving and protecting lives and relieving suffering [11]. In such medical emergencies seconds may save lives, so the emergency workers need to select the most appropriate services available in order to fulfill the incident requirements in the shortest time.

On the other hand the emergency service providers may change their status according to the availability of the services and resources which makes the decision making even more difficult due to unstable and impulsive pool of existing services.

In order to demonstrate the proposed scenario, all emergency service providers are supposed to provide their corresponding list of services publicly via appropriate RSS feeds. In the rest of this paper, different aspects of this scenario will be explored and the RSS-based approach for supporting the data integration and decision making process will be explained in details.

4 Semantic GeoRSS Data Integration

RDF Site Summary (RSS, also known as Really Simple Syndication) is an XML schema intended for description of content metadata and also content syndication. The RSS web feeds in its simplest form consists of a channel which contains multiple items. Each feed item is following a specific schema that comprises fields such as title, link to the content on the web, and the description of target content.

The latest specification of RSS (RSS 2.0.1) also provides an extension mechanism using XML namespaces that supports adding extra pieces of information to RSS feeds. This feature is especially important for the proposed solution of this paper, because it facilitates embedding of semantic information in RSS feeds. So for instance a GeoRSS feed, in addition to geographical information can also contain the semantic axioms which are connected to the domain ontologies by referring to the appropriate namespaces. Listing 1, demonstrates a GeoRSS data feed with embedded semantic annotations that describe the currently available services of the target hospital.

Please note that the standard namespaces such as RDF, OWL, and Geo as well as the namespace of domain ontology (service ontology) are defined in the header of RSS feeds. The RSS readers on the other hand, should be equipped with a semantic component that has access to domain ontologies in order to map the embedded semantics to the domain concepts and interpret the logical meaning of RSS items.

```
<?xml version="1.0"?>
<rss version="2.0"
      xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:service="http://www.accident.org/ontologies/Services.owl#">
  <channel>
     <title>St. Anna Hospital Updates</title>
     <description>
        latest service updates of St. Anna Hospital for the past 2 hours
    </description>
     <link>http://www.stanna.at</link>
     <dc:publisher>St. Anna Hospital</dc:publisher>
     <pubDate>Thu, 14 Jul 2011 23:56:15 PST</pubDate>

     <item>
       <pubDate> Thu, 14 Jul 2011 23:56:15 PST</pubDate>
       <title>Available services</title>
       <description>Available services for emergency cases</description>
       <link> http://www.stanna.at </link>
       <geo:lat>48.2161</geo:lat>
       <geo:long>16.3446</geo:long>

       <service:annotations>
          <owl:NamedIndividual rdf:about="&temporal;St.Anna">
               <rdf:type rdf:resource="&service;Childrenclinic"/>
               <service:hasService
                    rdf:resource="&service;surgeryRoomService"/>
               <service:hasService
                    rdf:resource="&service;surgeryTeamService"/>
            </owl:NamedIndividual>
       </service:annotations>

     </item>
   </channel>
 </rss>
```

Listing 1. A Semantic GeoRSS feed for description of hospital services

The client systems which are subscribed to the such data feeds can then read the latest status of corresponding services in real-time and update their solution space accordingly. As explained before the solution space may change quickly according to dynamic and impulsive nature of domain services. So the information integration process should be flexible enough to allow frequent updates to the solution space.

To clarify this, consider the hospital services in context of emergency management use-case. Since the hospital services and resources (human, room, etc) are frequently changing, the corresponding solution space can become outdated and inconsistent which consequently may lead to wrong decisions. In this regard, RSS feeds provide an easy mechanism for reading the latest information and status of services from service providers and updating the solution space on the fly.

The semantically enriched solution model may also benefit from advances of semantic web technologies such as semantic inference and automatic classification for capturing the domain knowledge and answering complex queries. For instance in case of emergency management use-case, the semantic restrictions can be used to automatically classify the medical centers under different service categories. Listing 2, demonstrates the restrictions of a service class (in this case SurgeryReadyCenter) based on domain concepts and their relationships:

$$\begin{aligned} &\mathtt{SurgeryReadyCenter} \equiv \\ &\qquad \exists \mathbf{hasService}.\mathtt{SurgeryRoomService} \sqcap \\ &\qquad \exists \mathbf{hasService}.\mathtt{SurgeryTeamService} \end{aligned}$$

Listing 2. A Semantic Restriction for a service class to identify matching medical centers

After applying the semantic restriction and running the reasoner, the inferred class hierarchy lists the matching medical centers under their corresponding service classes. In the above example, all medical centers that have a free surgery room and also a surgery team will be classified as a surgery-ready center. Later on, this information will be used for answering relevant service queries. As such, part of the domain expertise which is usually delivered by human user, can be replaced by a semantic inference. Figure 1, shows the explanation of automatic classification for a sample medical center.

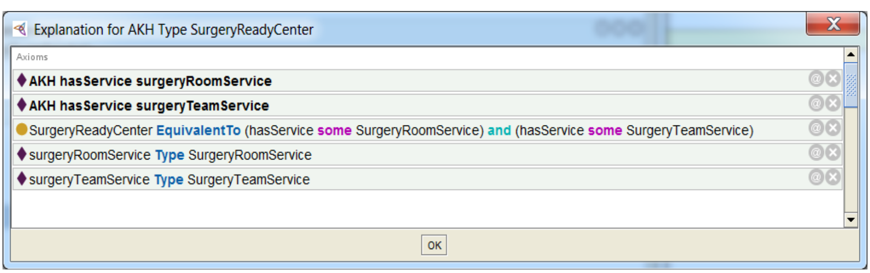


Fig. 1. The classification of medical centers under appropriate services after inference

Another important aspect of the proposed approach is the possibility to enrich the user requirements using Semantic Rules which is very helpful for scenarios such as emergency management use-case where first aid people perform the initial care before experts arrive. In such situations, the first aid person is usually able to see and report the symptoms but is not able to decide about required services and further medical processes. By proper definition of semantic rules on a well-defined ontology, new assertions about required services may facilitate the service query process. As an example, Listing 3, demonstrate a simple rule for identifying the required services for a possible concussion case.

```
@prefix accident: http://www.accident.org/ontologies/Accident.owl#
@prefix service: http://www.accident.org/ontologies/Services.owl#
@prefix temporal: http://www.accident.org/ontologies/Temporal.owl#

 [ concussion:
  (?person accident:hasSymptom accident:headache)
  (?person accident:hasSymptom accident:seizures)
  (?person accident:hasSymptom accident: unequalPupilSize)
  ->(?person service:requireService temporal:_BrainSurgeryReadyCenter)
 ]
```

Listing 3. A Rule for identifying the required services for a possible concussion case

Finally the semantic integrated model which is referred to as temporal model is comprised of the following components:

- A high-level service ontology that describes the basic concepts of service sink and service sources
- One or more domain ontologies that extend the service ontology and add the classification of domain services and relevant domain concepts. This ontology may also contain some defined classes for automatic classification of classes and instances
- Some semantic rules for enriching the user requirements based on the domain concepts
- The assertions about service providers and their attributes which are extracted from annotation part of Semantic GeoRSS feeds
- The service providers and their geographical location which can be extracted from GeoRSS feeds

As explained before the temporal model should be rebuilt regularly based on the latest information contained in GeoRSS feeds. After performing the automatic classification and applying the semantic rules, the temporal model is ready to answer queries regarding the appropriate service providers. The query results can be also further enriched and prioritized based on the nearest service provider location which is included in GeoRSS feeds. Figure 2, demonstrates the building blocks of temporal ontology and the corresponding information resources as semantically-enabled GeoRSS feeds.

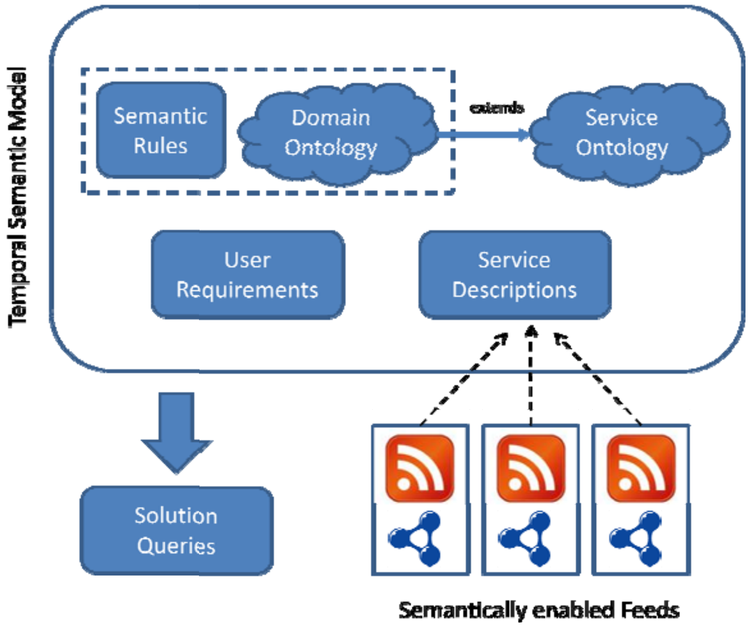


Fig. 2. The components of temporal ontology model

5 Conclusion and Future Works

In this paper a lightweight data integration approach for addressing the requirements of ad-hoc service discovery use-cases such as emergency management is presented. The proposed approach on one hand takes benefit of loosely coupled architecture of RSS feeds for data exchange and on the other hand uses the potential of Semantic Web technology for identifying the appropriate services. As proof of concept this methodology is applied to a set of imaginary medical centers that present their services via regularly published RSS feeds with embedded semantic information.

As future work we are intending to support an enterprise Mashup implementation via providing elaborated widgets for exploring the temporal ontology in a user-friendly way. Furthermore the same approach can be also used in some other domains such as e-government and business intelligence in order to support ad-hoc decision making and answering the relevant queries.

References

1. Anjomshoaa, A., Bader, G., Tjoa, A.M.: Exploiting Mashup Architecture in Business Use Cases. In: The 12th International Conference on Network-Based Information Systems (NBIS 2009), Indianapolis, USA (2009)
2. GeoRSS, `http://en.wikipedia.org/wiki/Georss` (last visited July 2011)

3. Tim O'Reilly, Levels of the Game: The Hierarchy of Web 2.0 Applications (July 2006), http://radar.oreilly.com/2006/07/levels-of-the-game-the-hierarc.html
4. Really Simple Syndication, http://en.wikipedia.org/wiki/RSS (last visited July 2011)
5. Representational State Transfer, Roy Thomas Fielding, Architectural Styles and the Design of Network-based Software Architectures, Ph.D. Thesis, University of California, Irvine, Irvine, California (2000)
6. IBM Mashup Center, http://www-01.ibm.com/software/info/mashup-center (last visited November 2009, last visited August 2011)
7. Content Creation and Distribution in an Expanding Internet Universe, White paper (May 2008), http://www.internetcontentsyndication.org/downloads/whitepapers/content_creation.pdf (last visited August 2011)
8. Udell, J.: RSS for Integration. InfoWorld (13), 40 (March 2005), http://www.infoworld.com/d/developer-world/using-rss-data-integration-283 (last visited August 2011)
9. OpenSearch, http://en.wikipedia.org/wiki/OpenSearch (last visited August 2011)
10. Tanasescu, V., Gugliotta, A., Domingue, J., Davies, R., Gutiérrez-Villarías, L., Rowlatt, M., Richardson, M., Stinčić, S.: A Semantic Web Services GIS Based Emergency Management Application. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 959–966. Springer, Heidelberg (2006)
11. Use Case Provenance of Decision Making Emergency Response, http://www.w3.org/2005/Incubator/prov/wiki/Use_Case_Provenance_of_Decision_Making_Emergency_Response (last visited August 2011)

ISDE 2011 PC Co-chairs' Message

Information systems in distributed environments (ISDE) is rapidly becoming a popular paradigm in the globalization era due to advancements in information and communication technologies. The distributed development of information systems as well as their deployment and operation in distributed environments impose new challenges for software organizations and can lead to business advantages. In distributed environments, business units collaborate across time zones, organizational boundaries, work cultures and geographical distances, which ultimately has led to an increasing diversification and growing complexity of cooperation among units. Successful participation in distributed environments, however, is ultimately a matter of the participants understanding and exploiting the particularities of their respective local contexts at specific points in time and exploring practical solutions through the local resources available.

The increased popularity of ISDE due to various factors has resulted in several research and industrial studies. Since information system development and implementation in distributed environments is still evolving and presents novel challenges, it is crucial to understand current research and practices in this regard and share results with researchers and practitioners in these areas.

The following selected papers of the ISDE 2011 International Workshop held in conjunction with the OTM conferences present recent advances and novel proposals in this direction.

In their paper "Towards a framework for Work Package Allocation for Global Software Development (GSD)," Marcos Ruano-Mayoral, Ricardo Colomo-Palacios, Joaquín M. Fernández-González, and Ángel García-Crespo report on the construction of a framework for work-package allocation within GSD projects. This framework lies on three main pillars: individual and organizational competency, organizational customization and sound assessment methods.

"Varying Levels of RFID Tag Ownership in Supply Chains" by Wei Zhou, Eun Jung Yoon, and Selwyn Piramuthu considers scenarios related to ownership of RFID tags in a supply chain, specifically the seamless incorporation of third-party logistics providers. The authors have developed a protocol for authentication of RFID-tagged items under such conditions.

"Software Quality Management Improvement Through Mentoring: An Exploratory Study from GSD Projects" by Ricardo Colomo-Palacios, Pedro Soto-Acosta, Alok Mishra, Ángel García-Crespo investigates whether mentoring, one of the lead personnel development tools, can improve the software quality management of projects developed under GSD.

José-Andrés Asensio, Luis Iribarne, Nicolás Padilla, and Cristina Vicente-Chicote, in their paper on "A Model-Driven Approach for Deploying Trading-Based Knowledge Representation Systems", present a model-driven engineering (MDE) approach aimed to help design and deploy trading-based knowledge representation (TKR) systems, a subset of management information systems (MIS).

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 178–179, 2011.
© Springer-Verlag Berlin Heidelberg 2011

"Communications in Global Software Development: An Empirical Study Using GTK+ OSS Repository" by Liguo Yu, Srini Ramaswamy, Alok Mishra, and Deepti Mishra presents an empirical study of the communication mechanisms in GNOME GTK+, a small-sized open-source distributed software project to identify how real-time and asynchronous communication methods could be used and balanced across global software development projects.

"Workflow Validation Framework in Distributed Engineering Environments" by Wikan Danar Sunindyo, Thomas Moser, Dietmar Winkler, Richard Mordinyi, and Stefan Biffl proposes the engineering service bus (EngSB) platform as a solution to integrate and validate heterogeneous workflows from different engineering fields and to link connections between different kinds of signals. In their paper on "Scalable Data Management in Distributed Information Systems," M. Remedios Pallardó-Lozoya, Javier Esparza-Peidro, José-Ramón García-Escrivá, Hendrik Decker, and Francesc D. Muñoz-Esco survey different techniques being used in modern SQL, NoSQL, and NewSQL systems in order to increase the scalability and adaptability in the management of persistent data for distributed applications managing dynamic workloads.

"Use Cases and Object Modeling Using ArgoUML" by Wojciech Complak, Adam Wojciechowski, Alok Mishra, and Deepti Mishra describes the process of creating guidelines and formal requirements for the design of a software system supporting storage and analysis of data concerning the regulatory duties of Polish Telecom in the range of the RIO framework agreement regarding collocation with the free UML modeling ArgoUML tool.

August 2011

Alok Mishra
Jürgen Münch
Ricardo Colomo-Palacios
Deepti Mishra

A Model-Driven Approach for Deploying Trading-Based Knowledge Representation Systems

José-Andrés Asensio[1], Luis Iribarne[1],
Nicolás Padilla[1], and Cristina Vicente-Chicote[2]

[1] Applied Computing Group, University of Almería, Spain
{jacortes,luis.iribarne,npadilla}@ual.es
[2] Department of Information and Communication Technologies
Technical University of Cartagena, Spain
cristina.vicente@upct.es

Abstract. Trading services are well-known solutions in *Software Engineering* for solving the interoperability and integration of software components and *Information Systems* (IS). This paper presents a *Model-Driven Engineering* (MDE) approach aimed to help designing and deploying *Trading-Based Knowledge Representation* (TKR) Systems, a subset of *Management Information Systems* (MIS). For this purpose, we have defined a set of modeling languages and supporting tools enabling: (a) the description of platform-independent TKR System architectures; (b) the high-level description of different deployment platforms; and (c) the creation of configuration models relating the two previous ones. These configuration models are then used as an input to a model-to-text transformation that generates the final TKR System implementation (code) from the selected platform. In order to demonstrate the feasibility and benefits of the proposed approach we also present a case study applied to an *Environmental Management Information System* (EMIS).

1 Introduction

Modern *Web-based Information Systems* (WIS) concern the use of many rules and standards for their deployment (and development), and which probably implies the use of: (a) a common vocabulary between all systems to support the knowledge representation of the system, providing probably the use of ontologies, and (b) a certain capacity to mediate objects, which allows the communication, negotiation, coordination, etc., between the system objects, i.e., interoperability solutions between software objects (perhaps services) of WIS. In this matter, current WIS are usually developed under open-distributed paradigms for globalization of information and the knowledge society, based on rules and standards of interaction and interconnection at runtime [11]. Trading services are well-known solutions in *Software Engineering* for solving the interoperability and the integration of software components and *Information Systems* [8]. Under this context, we have developed a model-driven extension of the traditional trading model

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 180–189, 2011.
© Springer-Verlag Berlin Heidelberg 2011

concept [5] as a solution to deploy knowledge-representation systems and to allow the interoperability and integration of software applications in WIS. In this paper we present a *Model-Driven Engineering* approach aimed to help designing and deploying *Trading-Based Knowledge Representation* (TKR) Systems. For this purpose, we have defined a set of modeling languages and supporting tools enabling: (a) the description of platform-independent TKR System architectures; (b) the high-level description of different deployment platforms; and (c) the creation of configuration models relating the two previous ones. These configuration models are then used as an input to a model-to-text transformation that generates the final TKR System implementation (code). Figure 1 summarizes this approach graphically, which will be described in the following.

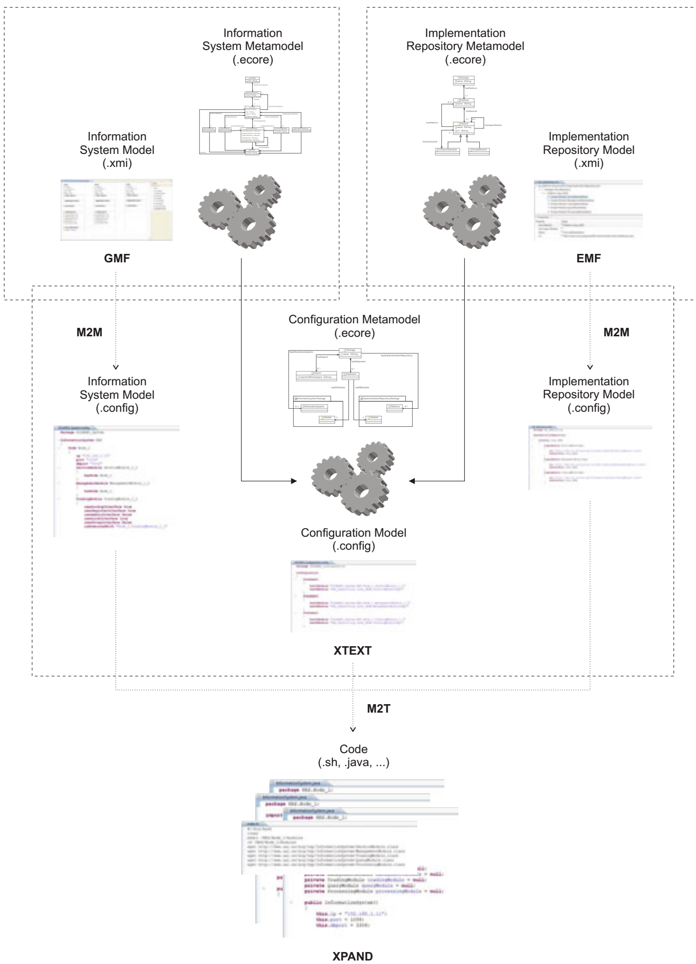


Fig. 1. Model-driven approach for the deployment of TKR systems

The modeling of platform-independent TKR system architectures appears in the upper left corner of the figure, and the modeling of the implementation platforms appears in the upper right corner. The modeling of the system configuration is shown relating the two previous ones in the middle of the figure. As a preliminary step, it is necessary to execute a *Model-To-Model* (M2M) transformation for both the information system and the implementation repository models to use them in the modeling of the system configuration. Finally, a *Model-To-Text* (M2T) transformation generates the final TKR system implementation for the selected platform from the configuration model. This approach integrates a wide range of Eclipse-based technologies such as EMF, GMF, *XText*, or *Xpand*, which are described throughout this paper. There are few works related to it in the literature. For instance, in [7] a methodology for configuring information systems that links BPM and xUML techniques is proposed and in [6] the authors provide a framework to enable the model-based development, simulation, and deployment of clinical information system prototypes.

In the following, the paper describes the proposal organized as follows: Section 2 presents the complete MDE approach for the design and deployment of TKR systems; Section 3 shows a case study which illustrates this approach; and, finally, Section 4 describes the conclusions and future work.

2 Description of the MDE-TKR Approach

This Section presents the MDE approach for the design and deployment of TKR systems that we propose. Firstly, it shows a metamodel for describing the architecture of these systems and another for describing a repository structure with the implementation of every module in the system. On the other hand, it presents an additional configuration metamodel for TKR systems, on which a language designed specifically to facilitate the creation of these configuration models is based; these models relate the architecture modules with their own implementation. Later, a M2T transformation, that lets to generate the necessary code to deploy the system in a real environment from these models, is described.

2.1 Trading-Based Knowledge Representation System Metamodel

The architecture of a TKR system can be distributed on different nodes. Each one must contain, at least, a *Service Module* (SM) —which provides a complete set of services shared by the other modules—, a *Management Module* (MM) —that acts as a link between the user interface and the other modules— and a *Querying Module* (QM) —concerned with the information queries demanded by the users—. Additionally, it can also contain zero or more *Trading Modules* (TM) —which enable the search and location of information in the system— and *Processing Modules* (PM) —responsible for managing the knowledge bases—. Compulsorily, the system must have, at least, a TM module and a PM module in some of its nodes. More than one instance of QM, TM and PM may be needed in order to accomplish the requirements of the system under design.

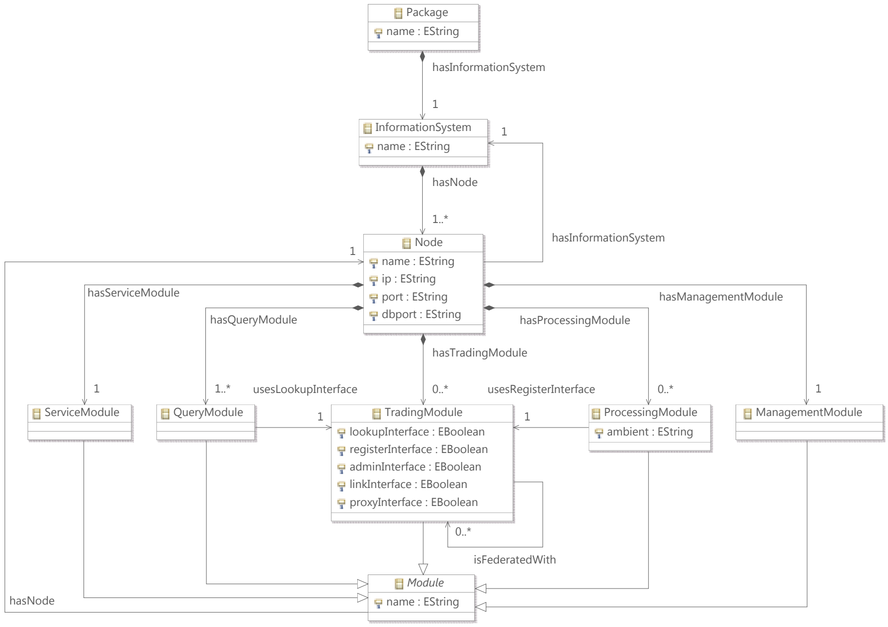


Fig. 2. TKR system metamodel

The metamodel proposed in Figure 2 has been defined using the *Eclipse Modeling Framework* (EMF) [3], which enables the description of TKR system models. It includes the concepts and restrictions previously described. It should also be noted the inclusion of the abstract *Module* metaclass with the *name* attribute, from which the different types of modules, previously described, inherit. The rest of the metaclasses already include the necessary properties (i.e., their attributes and relations with other metaclasses). For instance, the following attributes need to be set for each *Node*: an identifier (*name*), an IP address (*ip*), and two communication ports (*port* and *dbport*). Similarly, for each *TradingModule*, five *Boolean* attributes need to be set, each one indicating whether the trader implements or not the following interfaces: lookup (*lookupInterface*), re-gister (*registerInterface*), admin (*adminInterface*), link (*linkInterface*) and proxy (*proxyInterface*). Both the lookup and the register interfaces are required because each TM is inherently related to a metadata repository used for improving information retrieval [10]. The metamodel also enables the creation of TM federations (by means of the *isFederatedWith* relation) and the binding of each PM and QM (by means of the *usesRegisterInterface* and the *usesLookupInterface* relations, respectively) with the TM responsible of storing and querying the information in the metadata repository. In order to facilitate the creation of TKR system models, conforming to the metamodel described, we have implemented a graphical model editor using the Eclipse *Graphical Modelling Framework* (GMF) [4], that allow us to represent certain constraints, by using OCL rules [2].

2.2 Implementation Repository Metamodel

The structure of the implementation repository for TKR systems proposed in this paper consists of some deployment platforms (*Platform* metaclass) and the implementations of all the system modules in these platforms basically. As usual in *Software Engineering*, a module (*CompositeModule* metaclass) can be composed of one or more modules (*CompositeModule* or *SimpleModule*). Figure 3 (left) represents the implementation repository metamodel. The *Platform* metaclass includes a mandatory attribute (*name*) to specify the name of the deployment platform used. In addition, the abstract *Module* metaclass has been included, from which both composite and simple modules inherit, and two mandatory attributes, *name* and *uri*, have been defined to specify the module name and a reference to the location of the implementation files, respectively. For its part, each module has got a reference to its parent module and its platform. This metamodel has been defined using EMF too. The Eclipse reflective ecore model editor can be used for the creation of implementation repository models.

2.3 System Configuration Metamodel

Once defined the TKR system metamodel and the implementation repository metamodel, it is possible to relate each module in the system with its corresponding implementation so as to proceed to the final system deployment in the next step. For this reason, the *Statement* metaclass has been defined in the metamodel that Figure 3 shows (right), which only relates each module in the TKR system metamodel through its abstract *Module* metaclass with the abstract *Module* metaclass in the implementation repository metamodel. This metamodel also includes the *Import* metaclass with the *importedNamespace* attribute to enable its models import both TKR system models and implementation repository models. This metamodel has been defined using EMF.

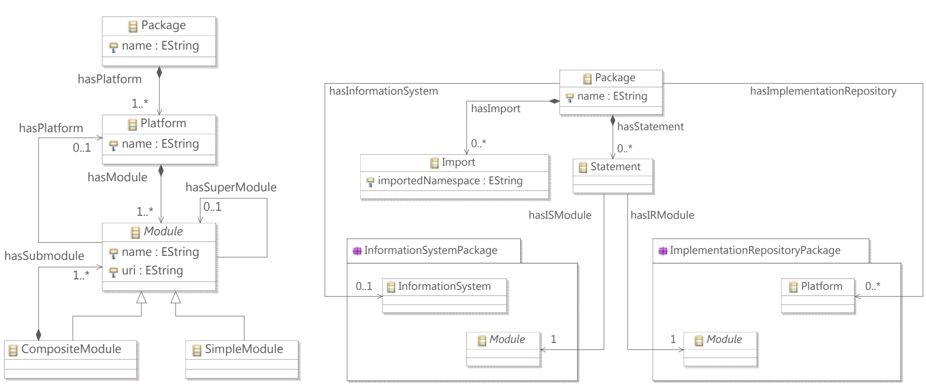


Fig. 3. Implementation repository and system configuration metamodels

2.4 System Configuration Language

To facilitate the creation of configuration models according to the metamodel described in Section 2.3, a *Domain-Specific Language* (DSL) has been implemented by using Eclipse *XText*. This tool allows to define a grammar from a metamodel, grammar that describes both the concrete syntax of the language that you want to implement and the procedure that the parser has to follow to generate a model.

Table 1 represents the part of the grammar defined for the configuration language of TKR systems. The grammar declaration appears in line #1, where its location and the reuse of org.eclipse.xtext.common.Terminals grammar — which defines rules for common terminals such as ID, STRING and INT— are indicated. From line #3 to #6 all ecore files with the metamodels used are declared. The rules to define the bi-directional mapping between the text and the model language appear in the following lines. The entry rule for the parser is in line #8. For instance, the rule defined from line #11 indicates that each *Node* entity begins with the "Node" keyword, followed by its name and the symbol "{". Then, their attributes are included with other keyword (i.e., "ip", "port", etc.), followed by the value of the attribute. Next, this entity defines the modules which compose itself and ends with the symbol "}".

An Eclipse editor is used to create the system configuration models. Previously, it is necessary to execute a M2M transformation for both the TKR system and the implementation repository models in order to can reference its elements from the configuration model.

Table 1. Definition of the system configuration grammar

```
1  grammar org.xtext.isconfig.config.Config with org.eclipse.xtext.common.Terminals
2
3  import "platform:/resource/ISConfig/metamodel/Configuration.ecore"
4  import "platform:/resource/ISConfig/metamodel/InfSystem.ecore" as InfSystemPackage
5  import "platform:/resource/ISConfig/metamodel/ImplRepository.ecore" as ImplRepPackage
6  import "http://www.eclipse.org/emf/2002/Ecore" as ecore
7
8  Package returns Package:
9    ...
10
11 Node returns InfSystemPackage::Node:
12   'Node' name=EString
13   '{'
14      'ip' ip=EString
15      'port' port=EString
16      'dbport' dbport=EString
17      hasServiceModule=ServiceModule
18      hasManagementModule=ManagementModule
19      ( hasTradingModule+=TradingModule (hasTradingModule+=TradingModule)* )?
20      hasQueryModule+=QueryModule (hasQueryModule+=QueryModule)*
21      ( hasProcessingModule+=ProcessingModule (hasProcessingModule+=ProcessingModule)* )?
22      'hasInfSystem' hasInfSystem=[InfSystemPackage::InformationSystem|EString]
23   '}';
24   ...
```

2.5 Code Generation

From the definition of a configuration model, it is possible to generate the necessary code for the deployment of the TKR system in a real environment which meets the requirements provided by the selected implementation platform. To carry out this process, a M2T transformation has been implemented with the help of the Eclipse *Xpand* tool, which enables code generation from a model created with the DSL editor. Table 2 represents the template defined for the transformation. The template describes the mapping between each entity in the model and its corresponding code. First sentence (line #1) is for the *Package* entity. This definition generates a script file (*make.sh*) with some commands, such as "clear" or "cd /", in line #3. Line #8 generates a Java file for each *Node* in the system, called *InformationSystem.java* and placed into a folder called as the value of the *name* attribute of the node, subfolder of another folder with the value of the *name* attribute of the information system. From line #9 the content of this file is declared. After executing the transformation the complete deployment of the TKR system occurs.

Table 2. Template for the M2T transformation by using *Xpand*

```
1  «DEFINE Package FOR ConfigurationPackage::Package»
2  «IF !this.hasStatement.isEmpty»
3    «FILE "make.sh"»
4      #!/bin/bash
5      clear
6      cd /
7      «FOREACH this.hasStatement.first().hasISModule.hasNode.hasInfSystem.hasNode AS node»
8      «FILE node.hasInformationSystem.name + "/" + node.name + "/InformationSystem.java"»
9       package «node.hasInformationSystem.name».«node.name»;
10      import «node.hasInformationSystem.name».«node.name».modules.*;
11      public class InformationSystem {
12        private String ip = null;
13        private int port = -1;
14        private int dbport = -1;
15        ...
```

3 Case Study: SOLERES-KRS

The model depicted in Figure 4 corresponds to the TKR subsystem of the SOLERES information system, an *Environmental Management* WIS [1], [9]. This model has been designed by using the GMF tool mentioned in Section 2.1. The subsystem is comprised of three nodes, all of them containing, in turn, the compulsory modules SM, MM and QM. In addition, the nodes with *id* "Node_1" and "Node_2" contain a PM and a TM, respectively, fulfilling the minimum system configuration requirements. The system contains a federation (dashed arrow) between the TMs included in "Node_1" and "Node_2", representing that the first module may delegate its information queries to the second one. Finally, the required links (solid arrows) between all PMs and QMs and the corresponding TMs (belonging the same or to other Nodes) have been also defined in the model.

The implementation repository has been modeled with the example of TKR system implementation by using the Java/JADE technologies. All the system functionality is embedded in software agents, for which it has been chosen the use of the JADE agent framework. Figure 5 represents this repository, where five simple modules have been specified for the *Platform Java/JADE*. This figure shows the properties for the *ServiceModuleImpl* too, among which appears an *uri* with the location of the implementation file (*ServiceModule.class*). Once defined these two models and executed the M2M transformation for both models to can reference their elements from a configuration model, the implemented DSL has been used to link each module included in the system with it corresponding implementation and thus define the configuration model of the subsystem SOLERES-KRS. This configuration model appears in Table 3.

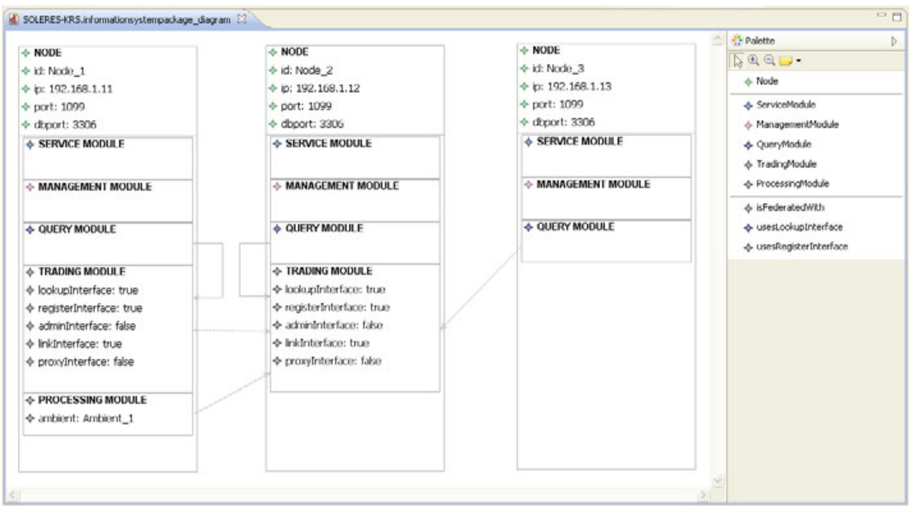


Fig. 4. SOLERES-KRS architecture model by using the GMF tool

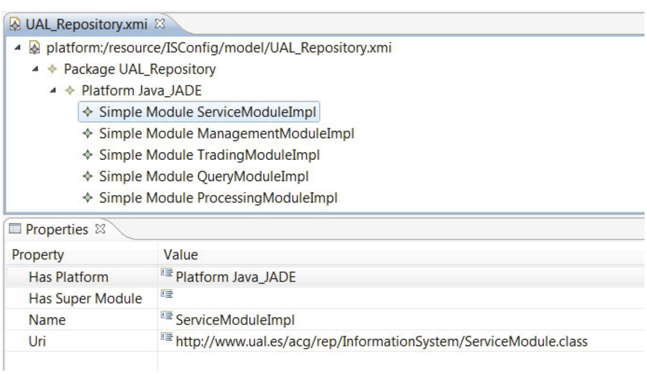


Fig. 5. Implementation repository model by using the reflective ecore model editor

Finally, the SOLERES-KRS deployment code for a real environment is generated from the definition of the configuration model and the M2T transformation presented in Section 2.5. This transformation generates several files: a script file which contains all the commands needed to get the implementation files of the system to be installated in a given environment (Table 4); and a set of files which makes both the configuration and initialization of all parameters from the values specified in the design of the system architecture (Table 5).

Table 3. SOLERES-KRS configuration model

```
Package SOLERES_Configuration
Configuration {
   Statement {
        hasISModule "SOLERES_System.KRS.Node_1.ServiceModule_1_1"
        hasIRModule "UAL_Repository.Java_JADE.ServiceModuleImpl" }
   Statement {
        hasISModule "SOLERES_System.KRS.Node_1.ManagementModule_1_1"
        hasIRModule "UAL_Repository.Java_JADE.ManagementModuleImpl" }
   Statement {
        hasISModule "SOLERES_System.KRS.Node_1.TradingModule_1_1"
        hasIRModule "UAL_Repository.Java_JADE.TradingModuleImpl" }
   ...
```

Table 4. Script file for the deployment of SOLERES-KRS

```
#!/bin/bash
clear
cd /
mkdir /KRS/Node_1/modules
cd /KRS/Node_1/modules
wget http://www.ual.es/acg/rep/InformationSystem/ServiceModule.class
wget http://www.ual.es/acg/rep/InformationSystem/ManagementModule.class
wget http://www.ual.es/acg/rep/InformationSystem/TradingModule.class
wget http://www.ual.es/acg/rep/InformationSystem/QueryModule.class
wget http://www.ual.es/acg/rep/InformationSystem/ProcessingModule.class
...
```

Table 5. Java file for the configuration of one of the nodes in SOLERES-KRS

```
package KRS.Node_1;
import KRS.Node_1.modules.*;
public class InformationSystem {
    private String ip = null;
    private int port = -1;
    private int dbport = -1;
    private ServiceModule serviceModule = null;
    private ManagementModule managementModule = null;
    private TradingModule tradingModule = null;
    private QueryModule queryModule = null;
    private ProcessingModule processingModule = null;
    public InformationSystem() {
      this.ip = "192.168.1.11";
      this.port = 1099;
      this.dbport = 3306;
      ...
```

4 Conclusions and Future Work

In this work we have presented a MDE approach to TKR system design and deployment in the context of WMIS. We have used three metamodels (a TKR system architecture metamodel, an implementation repository metamodel and a system configuration metamodel —with a DSL—, for relating the system architecture with its implementation), a M2M and a M2T transformations which aims to carry it out in a fast and easy way. This provides the separation and independence between the architecture and the platform, favouring the reutilization. Also, a case study of the SOLERES-KRS subsystem has been presented in order to illustrate the proposed approach. As for future work, one of the main lines that remains open in this work is the use of a trading system which makes the connection between the system architecture and the implementation repository models automatically, connection that currently a technician has to do manually using the DSL. Thus, the trading system would locate the best components to implement the TKR system without any intervention, favouring that the process could be carried out automatically from the design of the architecture with the Eclipse GMF tool.

Acknowledgment. This work has been supported by the EU (FEDER) and the Spanish Ministry MICINN under grant of the TIN2010-15588 and TRA2009-0309 projects, and the JUNTA de ANDALUCÍA under grant TIC-6114 project.

References

1. Asensio, J.A., Iribarne, L., Padilla, N., Ayala, R.: Implementing Trading Agents for Adaptable and Evolutive COTS Components Architectures. In: ICE-B 2008, Porto, Portugal, pp. 259–262 (2008)
2. Asensio, J.A., Iribarne, L., Padilla, N., Vicente, C.: A Trading-Based Knowledge Representation Metamodel for MIS Development. In: JISBD 2011 (in press, 2011)
3. EMF. Eclipse Modeling Framework, http://www.eclipse.org/modeling/emf
4. GMF. Graphical Modeling Framework, http://www.eclipse.org/modeling/gmf
5. Iribarne, L., Troya, J.M., Vallecillo, A.: A Trading Service for COTS Components. The Computer Journal 47(3), 342–357 (2004)
6. Mathe, J., Werner, I., Lee, Y., Malin, B., Ledeczi, A.: Model-based Design of Clinical Information Systems. Methods of Inf. in Medicine 47(5), 399–408 (2008)
7. Okawa, T., Kaminishi, T., Hirabayashi, S., Koizumi, H., Sawamoto, J.: An Information System Development Method Based on the Link of Business Process Modeling with Executable UML Modeling and its Evaluation by Prototyping. In: Workshops on Advanced Information Networking and Applications, vol. 1-3, pp. 1057–1064 (2008)
8. OMG. Trading Object Service Specification (2001), http://www.omg.org
9. Padilla, N., Iribarne, L., Asensio, J.A., Muñoz, F., Ayala, R.: Modelling an Environmental Knowledge-Representation System. In: Lytras, M.D., Damiani, E., Tennyson, R.D. (eds.) WSKS 2008. LNCS (LNAI), vol. 5288, pp. 70–78. Springer, Heidelberg (2008)
10. Trader, I.: ISO/IEC DIS 13235-1: IT–Open Distributed Processing–ODP Trading Function–Part 1: Specification (1996)
11. Xiao-feng, M., Bao-wen, X., Qing, L., Ge, Y., Jun-yi, S., Zheng-ding, L., Yan-xiang, H.: A Survey of Web Information Technology and Application. Wuhan University Journal of Natural Sciences 11(1), 1–5 (2006)

Software Quality Management Improvement through Mentoring: An Exploratory Study from GSD Projects

Ricardo Colomo-Palacios[1], Pedro Soto-Acosta[2], Alok Mishra[3], and Ángel García-Crespo[1]

[1] Computer Science Science Department, Universidad Carlos III de Madrid, Av. Universidad 30, Leganés, 28911, Madrid, Spain
{ricardo.colomo,angel.garcia}@uc3m.es
[2] Universidad de Murcia, Murcia, Spain, Campus de Espinardo, 30100 Espinardo, Murcia, Spain
psoto@um.es
[3] Department of Software Engineering, Atilim University, Incek, 06836, Ankara, Turkey
alok@atilim.edu.tr

Abstract. Software Quality Management (SQM) is a set of processes and procedures designed to assure the quality of software artifacts along with their development process. In an environment in which software development is evolving to a globalization, SQM is seen as one of its challenges. Global Software Development is a way to develop software across nations, continents, cultures and time zones. The aim of this paper is to detect if mentoring, one of the lead personnel development tools, can improve SQM of projects developed under GSD. The results obtained in the study reveal that the influence of mentoring on SQM is just temperate.

Keywords: Global Software Development, Mentoring, Software Quality Management.

1 Introduction

Distributed software development is becoming the norm for the software industry today [1]. GSD involves the development of application software through interactions of people, organizations, and technology across nations with different backgrounds, languages, and working styles [2]. GSD is a particular kind of Distributed Software Development (DSD) in which teams are distributed beyond the limits of a nation [3]. Cooperating over barriers of different organizations, nations, languages, time-zones and cultures is a multifaceted field of partially inter-related problems, including communication, knowledge exchange, and the coordination of international work groups [4].

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 190–199, 2011.
© Springer-Verlag Berlin Heidelberg 2011

This modern business strategy is based on developing high quality software in low-wage countries at low cost [5]. GSD has also been named as offshore software development, global software work, 24-hour development teams, follow the sun and round the clock.

Literature has reported several benefits related to the adoption of GSD. The most conveyed benefits include lower costs (e.g., [6]; [7]; [8]; [9]), greater availability of human resources and multi-skilled workforce (e.g. [10]; [11]; [12]; [13]), and shorter time-to-market cycles (e.g. [11], [14]; [15]). In a recent work [16], GSD is justified as because of the desire to extend working days, to benefit from the distribution of resources, to reduce costs and to be demographically closer to the target consumer.

But literature reported also challenges and issues related to GSD adoption. One of the challenges for GSD is quality and its management [12]. According to [17], quality usually is not directly affected by the distributed nature of GSD projects; however, some papers describe indirect effects of distributed collaboration on quality [18], [19]. Other authors are more categorical about the link between quality and GSD. Thus, [20] reported regular quality problems in the products developed offshore and [21] asseverates that the “follow the sun” model is essentially a quick-and-dirty strategy that converts a schedule problem into a quality disaster.

Given that quality management is an important competitive advantage in organizations with geographically distributed software development centers [22], the aim of this paper is to find out if mentoring could be an effective way to disseminate SQM practices among software development centers in order to mitigate the problems already reported in the literature.

The remainder of the paper is organized as follows. Section 2 surveys the relevant literature about mentoring. Section 3 describes the study about the use of mentoring as a facilitator of SQM in GSD environments. Section 4 brings the main conclusions and Section 5 depicts future works.

2 A Review of the Literature on Mentoring

Friday, Friday and Green [23] defined mentoring as an improvement process concerning a number of aspects related to a professional career, but also with the global improvement of the individual, which requires a senior advisor and a junior protégé. The People-Capability Maturity Model (P-CMM) [24] stated that the purpose of mentoring is to transfer the lessons learned from experienced personnel in a workforce competency to other individuals or workgroups. The pioneering work on mentoring [25], [26] suggested that mentoring is a powerful influence on success in organizational environments [27]. As a result of mentoring outputs, the protégé achieves a remarkable improvement in his professional career [28], [29], [30], a higher income [29], [31] and more satisfaction and social acceptance in the working environment [27], [32].

However, many recent studies reported that mentoring is a good predictor of an individual's career satisfaction yet only a very modest predictor of an individual's career ascendancy (e.g. [27], [31]). Thus, although mentoring mattered for career success, it represented just a part of a constellation of career resources that are embedded within individuals and their relationships [33].

Mentoring is a tool widely employed for knowledge management [34]. In software development projects, mentoring dramatically reduces the learning curve for inexperienced human resources [35], [36]. In this field, mentoring has been identified as a technique or strategy used for knowledge management [37] and human capital development [38]. Niazi et al. [39] pointed out that mentoring is a vital element of the implementation of software process improvement. More recently, mentoring has been identified as one of the leading success factors in adopting agile software development practices, since it expands the organizational culture [40], [41]. These results can be extended to expand national culture among foreign practitioners [42]. Nevertheless, in [43] authors stated the distance between the theoretical programme design and its application is one of the factors that decrease the efficiency of mentoring in software companies. In spite of their imperfect implementations, reports on the use of mentoring in GSD teams can be found (e.g. [44], [45], [46], [47], [48], [49]), however, to the best of author's knowledge, specific works about the influence of mentoring on SQM in GSD environments are still needed.

3 Study: Impact of Mentoring on SQM in GSD Scenarios

This section presents the study conducted in this paper. Such study is aimed to investigate the effects of mentoring techniques for SQM in the context of GSD working environments.

3.1 Research Design

This study is designed to be an exploratory study conducted using qualitative research techniques. The aim of the study is to identify which processes within SQM can be more influenced by the use of mentoring in GSD teams. Taking this into account, the output of the study is two-fold. The first output is a ranking of SQM processes with respect of its improvement by means of the application of mentoring. The second output is score on the impact of mentoring in such processes using a 1-4 Likert Scale (1= Low, 2= Medium, 3=High; 4=Very High).

The research tool selected to perform the study is a focus group. Focus groups involve assembling small groups of peers to discuss particular topics [50]. Discussion within these groups, although directed by a researcher, is largely free-flowing [51]. The use of discussion groups in software engineering and information systems development research activities has been extensively reported in the literature (e.g. [50], [51], [52], [53], [54], [55]).

Data collection was done as follows. The meeting was designed to be facilitated by three researchers (one in each location). Participants were connected using videoconference and assisted on-site by a researcher. The focus group's virtual meeting lasted approximately 35 minutes. During the meeting, researchers took extensive notes as well as videos. In accordance with previous literature [55], the session started with a brainstorming, where subjects thought about personal experiences on SQM, GSD and the use of mentoring. They use post-it notes to write down impressions and issues about the each of the SQM process. Once this step was completed, they discussed for 20 minutes the importance of each challenge and ranked the final list. The starting point was the list of processes of SQM. According to IEEE12207.0-96 [56], these processes are:

- Quality assurance process: The aim of this process is to provide assurance that the software products and processes in the project life cycle conform to their specified requirements by planning, enacting, and performing a set of activities to provide adequate confidence that quality is being built into the software [57].
- Verification process: Verification is an attempt to ensure that the product is built correctly, in the sense that the output products of an activity meet the specifications imposed on them in previous activities.
- Validation process: Validation is an attempt to ensure that the right product is built, that is, the product fulfils its specific intended purpose.
- Review process. Review is a process or meeting during which a software product is presented to project personnel, managers, users, customers, user representatives or other stakeholders for comment or approval. Reviews include Management reviews, Technical reviews, Inspections and Walk-throughs [58].
- Audit process. The purpose of a software audit is to provide an independent evaluation of the conformance of software products and processes to applicable regulations, standards, guidelines, plans, and procedures.

3.2 Sample Description

The sample consisted of one woman (20%) and four men (80%), with an average age of 42.4. Each of the participants was selected on the basis of his/her previous experience in all issues that the study covered: GSD, SQM and mentoring.

3.3 Results

Table 1 lists in alphabetical order the SQM processes explained by using excerpts from direct transcripts of the focus group session.

Table 1. Opinions of the influence of mentoring in SQM Processes within GSD projects

SQM Process	Excerpts
Audit process	Audits involve a formal group of independent people; it's not easy to suggest them to just one person" "Audits are expensive" "Audits are very difficult to assimilate by project personnel and company managers"
Quality assurance process	"I think that almost every software corporation has a software quality plan. But in the case of the absence of it, I think it won't be easy to convince them to adopt one" "Sometimes quality approach is not the same overseas" "This process can be seen as the key process here as it contains all the others in it. It's very broad"
Review process	"There are many kinds of reviews. Many of them rely on individuals. It's easier to convince an individual than a whole corporation" "I always suggest more junior professionals to perform managerial reviews. It does not matter if I'm performing a formal mentoring or just in an informal conversation"
Validation process	"Mentoring validation is the easiest thing here. It's easier to convince someone to look at requirements than to organize an audit, for instance"
Verification process	"I've bad and good experiences with verification and GSD. Talking about the bad ones, many times a partner presents a good quality plan but, once the development starts, there's not a single attempt to follow it"
General	"There are many differences among partners. There are several of them with high level of quality concern and others that their processes has nothing to do with quality"

Table 2 presents SQM processes ordered by importance and including the impact factor of mentoring among them. These results come from the sum of the punctuations given by subjects.

Table 2. Ordered SQM processes including mentoring impact factor

Rank	SQM Process	Mentoring Impact Factor
1	Validation process	2= Medium
2	Review process	2= Medium
3	Verification process	2= Medium
4	Quality assurance process	1= Low
5	Audit process	1= Low

3.4 Discussion

Results show that the impact of mentoring to adopt quality processes in GSD is restricted. This detail confronts with the fact that mentoring is a valid and recommended tool to implement quality related practices [59]. A possible explanation for this light influence may be the atomized analysis of SQM processes. This approach could be not convenient in our case. SQM can be adopted, but according to subjects' responses concrete cases and processes are not easy to mentor.

Thus, authors suggest combining mentoring with a companywide quality strategy in which norms and models must be adopted and updates to give quality to all software process. Concerning the importance of software improvement initiatives, subjects agree that, in many cases, the maturity of offshoring partners (CMMi maturity level) is higher than the contractor's.

Other important aspect regarding results is that subjects were informed that mentoring will be performed through the internet. As is widely reported in the literature, e-mentoring can be seen as the second best option, although it also has its advantages. A suggestion to improve the effectiveness of this technique could be to mix e-mentoring and t-mentoring in order to break the barriers of distance. However, due to the high cost of travelling, this set up must be considered only in long term relationships. Thus, suggestions could be to exchange software developers among project sites on a temporal basis in order to provide informal mentoring and cultural interchange; to organize workshops, especially at the beginning of the project and finally, to promote continuity in partnerships.

One issue that must be highlighted is the need of correct mentoring support. Both quality issues and mentoring success roots on the effective process support by the organization. Thus, organizations and managers must champion the process in terms of resources and times to aid the correct application of mentoring processes.

Finally, several works have highlighted the importance of quality issues for GSD (e.g. [60]) in terms of product quality and design quality, among others, but also claimed that there are not unique solutions to the known problems. Since GSD roots on cultural differences, on the construction of the third culture, mentoring can be a mean, but not the only way to develop it.

3.5 Limitations of the Study

The aim of this paper is to present an exploratory study. It may not be appropriate to generalize from a small sample (5 subjects) pertaining to European companies. However, taking into account that this is just a prospective study, data should provide potential start-points for further developments. In any case, the empirical research conducted is not strong enough to estimate the impact of mentoring on SQM in GSD.

For future works expanding the exploratory nature of this study authors suggest to expand both the sample and the composition of it in terms of corporations, nations and cultures represented.

Other important limitation comes from the level of granularity of the topics under study. Thus, V&V or audit processes might be too wide, for instance; the activity level could drive perhaps to more interesting conclusions.

4 Conclusions

This paper presents an exploratory study on the significance of mentoring practices for SQM in GSD teams. Results show that, although mentoring is seen as a good tool to support personnel development, the impact of these practices to mentor software quality issues is just moderate. However, the distribution of this impact among SQM practices is not equal. Processes like Validation, Review and Verification are more sensible to mentoring in GSD scenarios. This starting point gives way to a broader set of studies devoted to this issue. Next section depicts these studies.

5 Future Work

The current work proposes three types of initiatives which should be explored in future research. Firstly it is aimed to complement this exploratory and qualitative study with a more ambitious qualitative study along with a quantitative approach that enlarges the validity of the effort. The aim of the second study is to dig deeper into SQM processes and their best approach to implement them in GSD scenarios. Thus, it is aimed to get some measurable results of increased Software Quality due to the application of mentoring in GSD scenarios. Finally, authors propose to study the influence of other personnel development practices in this setup, more precisely, authors suggest to study the impact of coaching in the spread of SQM practices.

References

1. Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á., Soto-Acosta, P.: Team Software Process in GSD Teams: A study of new work practices and models. International Journal of Human Capital and Information Technology Professionals 1(3), 32–53 (2010)
2. Herbsleb, J.D., Mockus, A.: An empirical study of speed and communication in globally distributed software development. IEEE Transactions on Software Engineering 29(9), 481–494 (2003)
3. Herbsleb, J.D., Moitra, D.: Global software development. IEEE Software 18(2), 16–20 (2001)
4. King, W.R., Torkzadeth, G.: Information systems offshoring: research status and issues. MIS Quarterly 32(2), 205–225 (2008)
5. Khan, S.U., Niazi, M., Ahmad, R.: Factors influencing clients in the selection of offshore software outsourcing vendors: an exploratory study using a systematic literature review. Journal of Systems and Software 84(4), 686–699 (2011)
6. Smite, D., Wohlin, C., Gorschek, T., Feldt, R.: Empirical Evidence in Global Software Engineering: A Systematic Review. Journal of Empirical Software Engineering 15(1), 91–118 (2010)
7. García-Crespo, Á., Colomo-Palacios, R., Soto-Acosta, P., Ruano-Mayoral, M.: A Qualitative Study of Hard Decision Making in Managing Global Software Development Teams. Information Systems Management 27(3), 247–252 (2010)
8. Krishna, S., Sahay, S., Walsham, G.: Managing cross-cultural issues in global software outsourcing. Communications of the ACM 47(4), 62–66 (2004)
9. Ramasubbu, N., Krishnan, M.S., Kompalli, P.: Leveraging global resources: A process maturity framework for managing distributed development. IEEE Software 22(3), 80–86 (2005)
10. Conchuir, E.O., Holmstrom-Olson, H., Agerfalk, P.J., Fitzgerald, B.: Benefits of Global Software Development: Exploring the Unexplored. Software Process Improvement and Practice 14(4), 201–212 (2009)

11. Kommeren, R., Parviainen, P.: Philips experiences in global distributed software development. Empirical Software Engineering 12(6), 647–660 (2007)
12. Jiménez, M., Piattini, M., Vizcaíno, A.: Challenges and Improvements in Distributed Software Development: A Systematic Review. Advances in Software Engineering, Article ID 710971 (2009)
13. Milewski, A.E., Tremaine, M., Köbler, F., Egan, R., Zhang, S., O'Sullivan, P.: Guidelines for Effective Bridging in Global Software Engineering. Software Process: Improvement and Practice 13(6), 477–492 (2008)
14. Jalote, P., Jain, G.: Assigning tasks in a 24-hour software development model. Journal of Systems and Software 79(7), 904–911 (2006)
15. Sooraj, P., Mohapatra, P.K.J.: Modeling the 24-h software development process. Strategic Outsourcing: An International Journal 1(2), 122–141 (2008)
16. Palacio, R.R., Vizcaíno, A., Morán, A.L., González, V.M.: Tool to facilitate appropriate interaction in global software development. IET Software 5(2), 157–171 (2011)
17. Lamersdorf, A., Münch, J.: Studying the Impact of Global Software Development Characteristics on Project Goals: A Causal Model. The Open Software Engineering Journal 4, 2–13 (2010)
18. Sakthivel, S.: Managing Risks in Offshore Systems Development. Communications of the ACM 50(4), 69–75 (2007)
19. DeLone, W., Espinosa, J.A., Lee, G., Carmel, E.: Bridging global boundaries for IS project success. In: Proceedings of the 38th Hawaii International Conference on System Sciences (2005)
20. Jaakkola, H.: Towards a Globalized Software Industry. Acta Polytechnica Hungarica 6(5), 69–84 (2009)
21. Seshagiri, G.: GSD: not a business necessity but a of folly. IEEE Software 23(5), 63–64 (2006)
22. Sa, L., Marczak, S., Antunes, D., Audy, J.L.N.: Quality Management as a Competitive Strategy in a Distributed Software Development Environment. In: Proceedings of the Americas Conference on Information Systems Paper 208 (2003)
23. Friday, E., Friday, S.S., Green, A.L.: A reconceptualization of mentoring and sponsoring. Management Decision 42(5), 628–644 (2004)
24. Curtis, B., Hefley, W.E., Miller, S.A.: People Capability Maturity Model (P-CMM®) Version 2.0, Second Edition. CMU/SEI-2009-TR-003 (2009)
25. Levinson, D.J., Darrow, C.M., Klein, E.G., Levinson, M.H., McKee, B.: The Seasons of a Man's Life. Knopf, New York (1978)
26. Kram, K.E.: Phases of the mentoring relationship. Academy of Management Journal 26(4), 608–625 (1983)
27. Kammeyer-Mueller, J.D., Judge, T.A.: A quantitative review of mentoring research: Test of a model. Journal of Vocational Behavior 72(3), 269–283 (2008)
28. Blicke, G., Witzki, A.H., Schneider, P.B.: Mentoring support and power: A three year predictive field study on protégé networking and career success. Journal of Vocational Behavior 74(2), 181–189 (2009)
29. Dreher, G.F., Ash, R.A.: A comparative study of mentoring among men and women in managerial, professional, and technological positions. Journal of Applied Psychology 75, 539–546 (1990)
30. O'Brien, K.E., Biga, A., Kessler, S.R., Allen, T.A.: A meta-analytic investigation of gender differences in mentoring. Journal of Management 36(2), 537–554 (2008)
31. Blicke, G., Witzki, A.H., Schneider, P.B.: Self-initiated mentoring and career success: A predictive field study. Journal of Vocational Behavior 74(1), 94–101 (2009)

32. Chao, G.T., Walz, P.M., Gardner, P.D.: Formal and informal mentorships: A comparison on mentoring functions and contrast with non mentored counterparts. Personnel Psychology 45, 619–636 (1992)
33. Singh, R., Ragins, B.R., Tharenou, P.: What matters most? The relative role of mentoring and career capital in career success. Journal of Vocational Behavior 75(1), 56–67 (2009)
34. Abdul-Rahman, H., Wang, C.: Preliminary approach to improve knowledge management in engineering management. Scientific Research and Essays 5(15), 1950–1964 (2010)
35. Ramaswamy, R.: Mentoring object-oriented projects. IEEE Software 18(3), 36–40 (2001)
36. Lesser, E.L., Storck, J.: Communities of practice and organizational performance. IBM Systems Journal 40(4), 831–841 (2010)
37. Fehér, P., Gábor, A.: The role of knowledge management supporters in software development companies. Journal of Software Process Improvements and Practice 11(2), 251–260 (2006)
38. Soto-Acosta, P., Casado-Lumbreras, C., Cabezas-Isla, F.: Shaping human capital in software development teams: the case of mentoring enabled by semantics. IET Software 4(6), 445–452 (2010)
39. Niazi, M., Wilson, D., Zowghi, D.: Critical success factors for software process improvement implementation: an empirical study. Journal of Software Process Improvements and Practice 11(2), 193–211 (2006)
40. Misra, S.C., Kumar, V., Kumar, U.: Identifying some important success factors in adopting agile software development practices. Journal of Systems and Software 82(11), 1869–1890 (2009)
41. Shih, C.C., Huang, S.J.: Exploring the relationship between organizational culture and software process improvement deployment. Information & Management 47(5-6), 271–281 (2010)
42. Casado-Lumbreras, C., Colomo-Palacios, R., Soto-Acosta, P., Misra, S.: Culture dimensions in software development industry: The effects of mentoring. Scientific Research and Essays 6(11), 2403–2412 (2011)
43. Casado-Lumbreras, C., Colomo-Palacios, R., Gómez-Berbís, J.M., García-Crespo, Á.: Mentoring programmes: a study of the Spanish software industry. International Journal of Learning and Intellectual Capital 6(3), 293–302 (2009)
44. Oshri, I., Kotlarsky, J., Willcocks, L.P.: Global Software Development: Exploring socialization in distributed strategic projects. Journal of Strategic Information Systems 16(1), 25–49 (2007)
45. Colomo-Palacios, R., Mishra, A., García-Crespo, A., Ruano-Mayoral, M.: Mentoring in Global Software Development Teams. In: Proceedings of the 20th Annual Global Information Technology Management Association World Conference (2011)
46. Casey, V., Richardson, I.: Virtual Teams: Understanding the Impact of Fear. Software Process: Improvement and Practice 13(6), 511–526 (2008)
47. Casey, V., Richardson, I.: Implementation of Global Software Development: A Structured Approach. Software Process: Improvement and Practice 14(5), 247–262 (2009)
48. Lacity, M.C., Rottman, J.W.: Effects of offshore outsourcing of information technology work on client project management. Strategic Outsourcing: An International Journal 2(1), 4–26 (2009)
49. Suchan, J., Hayzak, G.: The Communication Characteristics of Virtual Teams: A Case Study. IEEE Transactions on Professional Communication 44(3), 174–186 (2001)
50. Baddoo, N., Hall, T.: Motivators of Software Process Improvement: an analysis of practitioners' views. The Journal of Systems and Software 62(2), 85–96 (2002)

51. Hall, T., Beecham, S., Rainer, A.: Requirements problems in twelve software companies: an empirical analysis. IEE Proceedings 149(5), 153–160 (2002)
52. Beecham, S., Hall, T., Rainer, A.: Defining a Requirements Process Improvement Model. Software Quality Journal 13(3), 247–279 (2005)
53. Bennett, C., Myers, D., Storey, M.-A., German, D.M., Ouellet, D., Salois, M., Charland, P.: A survey and evaluation of tool features for understanding reverse engineered sequence diagrams. Journal of Software Maintenance and Evolution: Research and Practice 20(4), 291–315 (2008)
54. Casey, V.: Virtual software team project management. Journal of the Brazilian Computer Society 16(2), 83–96 (2010)
55. Karlsson, L., Dahlstedt, A.G., Regnell, B., Natt Och Dag, J., Persson, A.: Requirements engineering challenges in market-driven software development – An interview study with practitioners. Information and Software Technology 49(6), 588–604 (2007)
56. IEEE/EIA 12207.0-1996//ISO/IEC12207:1995, Industry Implementation of Int. Std ISO/IEC 12207:95, Standard for Information Technology-Software Life Cycle Processes. IEEE (1996)
57. Abran, A., Moore, J.W. (eds.): Guide to the Software Engineering Body of Knowledge: 2004 Version. IEEE Computer Society (2004)
58. IEEE Computer Society, IEEE Standard for Software Reviews, IEEE Std 1028-1997. IEEE: New York, NY, USA (1998)
59. Schneidewind, N.F.: Body of Knowledge for Software Quality Measurement. IEEE Software 35(2), 77–83 (2002)
60. Yu, L., Mishra, A.: Risk Analysis of Global Software Development and Proposed Solutions. Automatika 51(1), 89–98 (2010)

Towards a Framework for Work Package Allocation for GSD

Marcos Ruano-Mayoral[1], Ricardo Colomo-Palacios[2], Joaquín M. Fernández-González[1], and Ángel García-Crespo[2]

[1] EgeoIT
Av. Brasil 17, 28020 Madrid, Spain
{marcos.ruano,joaquin.fernandez}@egeoit.com
[2] Computer Science Science Department, Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, Spain
{ricardo.colomo,angel.garcia}@uc3m.es

Abstract. Global software development is an inexorable trend in the software industry. The impact of the trend in conventional software development can be found in many of its aspects. One of them is task or work package allocation. Task allocation was traditionally driven by resource competency and availability but GSD introduces new complexities to this process including time-zones differences, costs and cultural differences. In this work a report on the construction of a framework for work-package allocation within GSD projects is presented. This framework lies on three main pillars: individual and organizational competency, organizational customization and sound assessment methods.

Keywords: Global Software Development, task allocation, competency, cultural differences.

1 Introduction

Globalization in business is a trend that is here to stay. One of the results of globalization is the transfer of several production processes from one country to other. In software industry, this transfer is known as Distributed Software Development (DSD). The nature of the software production process enables its globalization and as a result of this, today, it is common for software development to take place in multiple or even distributed groups working together on a common system [1]. DSD has been facilitated by the revolution in communication which the Internet has made possible, developing tools that made feasible communication among development sites [2].

Global Software Development (GSD) can be seen as a particularization of DSD. Not in vain, according to [3], GSD is a particular kind of DSD in which teams are distributed beyond the limits of a given nation. More precisely, when the distance becomes global in DSD, with team members distributed around the world, this characterizes GSD [4]. Approaches have ranged from subcontracting portions of software development projects to third-party companies or subsidiaries, to establishing truly virtual development teams [5].

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 200–207, 2011.
© Springer-Verlag Berlin Heidelberg 2011

GSD gained momentum as it promised spectacular benefits but also attracted attention due to the complexity and challenges related to GSD [6]. Compared to localized software development, global software development has various risks, such as those induced by the spatial and temporal distance between development teams [7]. Literature presents many works devoted to underline GSD benefits (e.g. [8]) and caveats (e.g. [9]). In spite of the challenges, an increasing amount of software projects are run under GSD and this practice is becoming a norm in the software industry [10].

Because of its popularity, GSD must be studied carefully. However, the art and science of global software development is still evolving [10] and thus, immature [6].

Given that work allocation is an important issue in GSD [11], the aim of this paper is to define a framework to help managers to distribute work packages among project partners in GSD projects.

The remainder of the paper is organized as follows. Section 2 surveys the relevant literature about task allocation in GSD. Section 3 describes the framework to allocate work packages among project partners. Section 4 brings the main conclusions and future works.

2 Task Allocation in GSD

Traditionally, the allocation process of personnel within software development projects is mainly based on the technical skills of the teams and the task requirements [12]. In GSD scenario, task allocation is quite different, involving aspects like costs, time-zone differences and cultural imparities among others. As a result of this diversity, the allocation of tasks and responsibilities to distributed teams can have a significant impact on GSD project success [13]. While, back in 2008, [11] stated that there is a lack of attention to the problem of allocating projects in distributed teams, recent and relevant efforts have been devoted to this field of study.

Maybe the most relevant efforts come from the works of Lamersdorf, Munch et al. [14], [15], [16], [17], [18]. In these works, authors overview the different kinds of GSD projects and propose a set of criteria that is used when an allocation of tasks in a GSD project is faced. Using this, these authors propose an approach to perform a systematic task allocation evaluation.

Other works have been devoted to support GSD task allocation using multi-criteria decision analysis [11], allocating resources for software product line projects [19], using work breakdown techniques [20], system dynamics approaches [21] or just basic module allocation [22].

In this work a different approach is adopted. By means of qualitative research, authors propose a framework for work package allocation. This approach is novel, due to three intrinsic characteristics. Firstly, its novelty roots on the organizational and personal competency orientation of the framework. In the second term, its originality is based on the customization that this approach permits and needs to be fully operative. In [14], adaptability was one of the requirements of the tool developed. But apart from this adaptability, it is needed to describe the approach to achieve this adaptability. This is the second big contribution of the work. Finally, the framework is based on the use of state of the art assessment methods. Thus, instead of a set of standard assessment metrics (using, for instance Likert-like values), the approach is to provide a set of sound and different techniques to evaluate all variables present in GSD task and work-package allocation.

3 A Framework for Work Package Allocation in GSD

This section describes the framework designed to assist work package allocation among GSD partners and locations.

3.1 Framework Construction Process

The framework development consists in two main phases, as depicted in Figure 1.

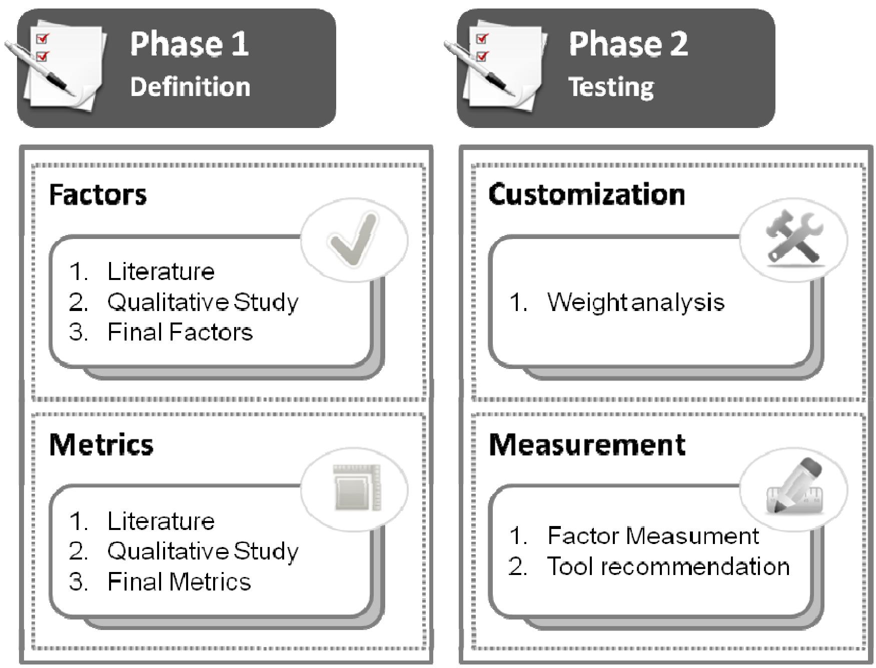


Fig. 1. Definition and testing phases of the framework

Phase 1 comprehends the definition of the framework. This definition presents two main steps. The first one is the definition of the factors that are present in the allocation of resources in GSD projects. This definition is performed using a three point approach. Firstly, a study of the literature is performed. The output of this study is the set of initial factors that must be taken into account to allocate work-packages in GSD scenarios. The second point is a qualitative study in which several experts are gathered using the focus group technique. In this case, the output of this study is a refined list of factors. The Focus Group method involves assembling small groups of peers to discuss particular topics [23]. The discussion within these groups, although directed by a researcher, is largely free-flowing [24]. Finally, authors compile a final list of factors is developed as the final output of Phase 1 Step 1. The second step is aimed to specify a set of valid metrics for all elements defined in Step 1. To do so, in the first term, an analysis of the literature is performed. Later, a qualitative study is performed in order to set a single metric for each factor. In this case, Delphi Study is the tool to achieve this objective. Delphi method was designed by Dalkey and Helmer [25] in the 1950s for military purposes and, from the 1960s onwards it was also used in organizational spheres thanks to the further developments of these authors [26]. This method uses a panel of experts as the source of information and questionnaires as the medium of interaction [27]. The Delphi method presents three main features [28]: anonymous response; iteration and controlled feedback; and statistical group response. Once the metrics are selected, this step ends with the definition and documentation of such metrics.

Phase 2 represents the customization of results obtained in Phase 1 to a particular GSD environment. This phase presents two steps. In Step 1, given a particular GSD project, all factors are weighted in order to adapt the decision to the expectatives and aims of the participants. Phase 2 Step 1 must be completed using techniques like interviews and qualitative approaches to get a common and agreed response to the weights of the project. In Step 2 all factors are measured using the metrics defined in Phase 1 Step 2. Once all measures have been obtained, by means of an automated tool, a set of recommendations for work-package allocation are issued.

In complex projects where many partners and work-packages are involved, different configurations cannot be handled without the help of an automated tool. WP-Locate is the proof-of-the-concept tool to support this decision making process. This tool stores three different kinds of information: project requisites (e.g. minimum assignations, round the clock need), work-package descriptions mainly in terms of competency needs and interactions and partners' information (competency levels and other variables). WP-Locate, taking into account the restrictions of the given project, provides a set of possible assignations ranked according to the assessment of every factor included in it.

3.2 Factors: Initial Steps

According to a preliminary study, the work of Lamersdorf, Münch and Rombach [17] identifies almost all factors that should appear in task or work-package allocation within GSD projects. These factors are:

- Cost
- Time
- Temporal Distance
- Geographic Distance
- Cultural Distance
- Size
- Economic situation

- Customer Proximity
- Collaboration history
- Strategic Considerations
- Software development maturity
- Human Resource Management maturity
- Trust
- Competency.

Focusing on competency in the individual and organizational model, the aim of this framework is to integrate competency measurement in these two levels in order to integrate in these two broad dimensions many of the aspects found. Based in previous works of the authors ([29], [30], [31], [32], [33]), the aim here is to integrate competence definition and assessment with the final objective to get a matching between the required competences and skills and those available in a given GSD site. This must be done comparing competency descriptions in the individual level and competence needs of each work-package.

With regard to cultural differences, authors believe that, although these factors are widely present in the software development related literature (e.g. [34], [35], [36], [37], [38], [39]), this are not properly included in decision making processes. Thus, the aim is to integrate findings present in the works of Hofstede (e.g. [40], [41 [42]) in the assessment of cultural differences by means of the comparison of the five cultural dimensions reported by this author, namely Power distance, Individualism / collectivism, Masculinity / femininity, Uncertainty avoidance and Long-term / short-term time orientation.

3.3 Testing Environment

The aim of PROPS-Tour project is to design, build and prototype an ICT platform for open and integrated promotion of different tourism services available in a certain geographic area, ensuring an efficient and unbiased provision of services information. PROPS-Tour is conceived as a catalyst to allow an uneven and heterogenic tourist services offering to reach its natural targets: tourists already in their destination. The promotion mechanisms will therefore allow the elaboration of updated and consistent information about the plethora of tourism services to feed recommendation systems that tourist may already use. The main technological and functional innovations provided by the PROPS-Tour project can be summed up as:

- Construction of a platform capable of thoroughly storing all the tourist services offered by the different providers in a certain destination regardless their size, presence or technological capacity.
- Recording the complete descriptions about services as well as the required semantic annotation for subsequent processing.
- Creation of a social interface so that services providers are able to refresh and update those contents considered as highly valuable for its immediacy.
- Provision of the required interfaces to allow the communication between the repository and other recommendation systems by means of open standards.
- Development of the necessary mechanisms so that the platform is able to promote equally all the recorded services generating no biases of any kind.

- Elaboration of metrics focused on the platform impact and relevance in the small tourism companies' ecosystem, identifying their influence in productivity and sustainability.

The project has been endorsed by the Eureka Network with the E! Label (E! 6244) and its development will involve a consortium comprehending four different partners from two European countries. In order to achieve the objectives of the project and to fulfill the requirements of the Eureka Programme, especially those regarding the effective cooperation and collaboration among the partners, the project plan was structured into tasks, following the PSS-05-0 standard for software development projects (provided by the European Space Agency), and those tasks where grouped into work packages with different functional or methodological scope. Additionally, the consortium is currently considering the possibility of subcontracting some of the tasks in the project. Thus, the project can be considered as a GSD project and is suitable to be impacted by the framework introduced in this paper in terms of the support to the decision making regarding the allocation of the work packages and tasks of the project to the partners and subcontractors.

4 Conclusions and Future Work

This paper describes the ongoing research performed towards the definition of a framework for the allocation of work-packages among project partners in a GSD project set. The aim of the framework is to integrate sound assessment methods to common allocation factors in order to provide a set of ranked allocation recommendations aimed to fulfill partners' needs and maximize resources and effort.

This work is heading towards a three-pronged approach in terms of future work. The first is the full definition and testing of this framework. The second is the expansion of this framework to include improvement and organizational learning issues. Finally, it is aimed to integrate semantics in the solution by means of the adoption of competency ontologies and other formalisms to provide web 3.0 like machine processable content.

Acknowledgments. This work is supported by the Spanish *Centro para el Desarrollo Tecnológico Industrial* (CDTI) under the Eureka Project E! 6244 PROPS-Tour and the national cooperation project SEM-IDi (IDI-20091150).

References

1. Ovaska, P., Rossi, M., Marttiin, P.: Architecture as a Coordination Tool in Multi-site Software Development. Software Process: Improvement and Practice 8(4), 233–247 (2003)
2. Casey, V.: Developing Trust In Virtual Software Development Teams. Journal of Theoretical and Applied Electronic Commerce Research 5(2), 41–58 (2010)
3. Herbsleb, J.D., Moitra, D.: Global software development. IEEE Software 18(2), 16–20 (2001)
4. Prikladnicki, R., Audy, J.L.N.: Process models in the practice of distributed software development: A systematic review of the literature. Information and Software Technology 52(8), 779–791 (2010)

5. Kobitzsch, W., Rombach, D., Feldmann, R.L.: Outsourcing in India. IEEE Software 18(2), 78–86 (2001)
6. Smite, D., Wohlin, C., Gorschek, T., Feldt, R.: Empirical evidence in global software engineering: a systematic review. Empirical Software Engineering 15(1), 91–118 (2010)
7. Yu, L., Mishra, A.: Risk Analysis of Global Software Development and Proposed Solutions. Automatika 51(1), 89–98 (2010)
8. Conchuir, E.O., Holmstrom-Olson, H., Agerfalk, P.J., Fitzgerald, B.: Benefits of Global Software Development: Exploring the Unexplored. Software Process Improvement and Practice 14(4), 201–212 (2009)
9. Jiménez, M., Piattini, M., Vizcaíno, A.: Challenges and Improvements in Distributed Software Development: A Systematic Review. Advances in Software Engineering, Article ID 710971 (2009)
10. Damian, D., Moitra, D.: Global software development: How far have we come? IEEE Software 23(5), 17–19 (2006)
11. Barcus, A., Montibeller, G.: Supporting the allocation of software developmentwork in distributed teams with multi-criteria decision analysis. Omega 36, 464–475 (2008)
12. Duggan, J., Byrne, J., Lyons, G.J.: A Task Allocation Optimizer for Software Construction. IEEE Software 21(3), 76–82 (2004)
13. Noll, J., Beecham, S., Richardson, I.: Global software development and collaboration: barriers and solutions. ACM Inroads 1(3), 66–78 (2010)
14. Lamersdorf, A., Münch, J.: A multi-criteria distribution model for global software development projects. Journal of the Brazilian Computer Society 16(2), 1–19 (2010)
15. Lamersdorf, A., Münch, J., Rombach, D.: Towards a multi-criteria development distribution model: An analysis of existing task distribution approaches. In: ICGSE 2008: Proceedings of the 2008 IEEE International Conference on Global Software Engineering, pp. 109–118. IEEE Computer Society, Washington, DC (2008)
16. Lamersdorf, A., Münch, J., Rombach, D.: A Decision Model for Supporting Task Allocation Processes in Global Software Development. In: Bomarius, F., Oivo, M., Jaring, P., Abrahamsson, P. (eds.) PROFES 2009. LNBIP, vol. 32, pp. 332–346. Springer, Heidelberg (2009)
17. Lamersdorf, A., Munch, J., Rombach, D.: A survey on the state of the practice in distributed software development: Criteria for task allocation. In: Fourth IEEE International Conference on Global Software Engineering, ICGSE 2009, pp. 41–50 (2009)
18. Lamersdorf, A., Münch, J.: Model-based Task Allocation in Distributed Software Development. In: Nordio, M., Joseph, M., Meyer, B., Terekhov, A. (eds.) SEAFOOD 2010. LNBIP, vol. 54, pp. 37–53. Springer, Heidelberg (2010)
19. Pereira, T.A.B., dos Santos, V.S., Ribeiro, B.L., Elias, G.: A Recommendation Framework for Allocating Global Software Teams in Software Product Line Projects. In: Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering (2010)
20. Mohan, S., Fernandez, J.: New Opportunities presented by novel Work Breakdown techniques for Distributed Software Development. In: Proceedings of the International Conference on Global Software Engineering, pp. 305–307 (2010)
21. Setamanit, S.-o., Wakeland, W., Raffo, D.: Improving global software development project performance using simulation. In: Portland International Center for Management of Engineering and Technology, pp. 2458–2466 (2007)
22. Battin, R., Crocker, R., Kreidler, J., Subramanian, K.: Leveraging resources in global software development. IEEE Software 18(2), 70–77 (2001)
23. Baddoo, N., Hall, T.: Motivators of Software Process Improvement: an analysis of practitioners' views. The Journal of Systems and Software 62(2), 85–96 (2002)

24. Hall, T., Beecham, S., Rainer, A.: Requirements problems in twelve software companies: an empirical analysis. IEE Proceedings 149(5), 153–160 (2002)
25. Dalkey, N., Helmer, O.: The use of experts for the estimation of bombing requirements. A project Delphi experiment. The Rand Corporation, RM-727-PR (1951)
26. Dalkey, N., Helmer, O.: An experimental application of the Delphi method to the use of experts. Management Science 9(3), 458–467 (1963)
27. Blind, K., Cuhls, K., Grupp, H.: Personal attitudes in the assessment of the future of science and technology: a factor analysis approach. Technological Forecasting and Social Change 68(2), 131–149 (2001)
28. Landeta, J.: Current validity of the Delphi method in social sciences. Technological Forecasting & Social Change 73(5), 467–482 (2006)
29. Colomo-Palacios, R., García-Crespo, Á., Soto-Acosta, P., Ruano-Mayoral, M., Jiménez-López, D.: A case analysis of semantic technologies for R&D intermediation information management. International Journal of Information Management 30(5), 465–469 (2010)
30. Colomo-Palacios, R., Ruano-Mayoral, M., Soto-Acosta, P., García-Crespo, Á.: The War for Talent: Identifying competences in IT Professionals through semantics. International Journal of Sociotechnology and Knowledge Development 2(3), 26–36 (2010)
31. Colomo Palacios, R., García Crespo, A., Gómez Berbís, J.M., Casado-Lumbreras, C., Soto-Acosta, P.: SemCASS: technical competence assessment within software development teams enabled by semantics. International Journal of Social and Humanistic Computing 1(3), 232–245 (2010)
32. Colomo Palacios, R., Tovar Caro, E., García Crespo, A., Gómez Berbís, J.M.: Identifying Technical Competences of IT Professionals: The Case of Software Engineers. International Journal of Human Capital and Information Technology Professionals 1(1), 31–43 (2010)
33. Colomo Palacios, R., Ruano Mayoral, M., Gómez Berbís, J.M., García Crespo, A.: Semantic Competence Pull: A Semantics-Based Architecture for Filling Competency Gaps in Organizations. In: García, R. (ed.) Semantic Web for Business: Cases and Applications. IGI Global, Hershey (2008)
34. Casey, V.: Imparting the importance of culture to global software development. ACM Inroads 1(3), 51–57 (2010)
35. Shih, C.C., Huang, S.J.: Exploring the relationship between organizational culture and software process improvement deployment. Information & Management 47(5-6), 271–281 (2010)
36. Jaakola, H., Heimburger, A., Linna, P.: Knowledge-oriented software engineering process in a multi-cultural context. Software Quality Journal 18(2), 299–319 (2010)
37. Hahn, E.D., Bunyaratavej, K.: Services cultural alignment in offshoring: The impact of cultural dimensions on offshoring location choices. Journal of Operations Management 28(3), 186 (2010)
38. Aramo-Immonen, H., Jaakkola, H., Keto, H.: Multicultural Software Development: The Productivity Perspective. International Journal of Information Technology Project Management 2(1), 19–36 (2011)
39. Casado-Lumbreras, C., Colomo-Palacios, R., Soto-Acosta, P., Misra, S.: Culture dimensions in software development industry: The effects of mentoring. Scientific Research and Essays 6(11), 2403–2412 (2011)
40. Hofstede, G.H.: Culture's Consequences: International Differences in Work-Related Values. Sage Publications, Newbury Park (1984)
41. Hofstede, G.H.: Culture's Consequences: Comparing Values, Behaviours, Institutions, and Organisations, 2nd edn. Sage Publications, Thousand Oaks (2001)
42. Hofstede, G.H.: Cultures and Organizations: Software of the Mind. McGraw-Hill (2010)

Scalable Data Management in Distributed Information Systems

M. Remedios Pallardó-Lozoya, Javier Esparza-Peidro,
José-Ramón García-Escrivá, Hendrik Decker, and Francesc D. Muñoz-Escoí

Instituto Universitario Mixto Tecnológico de Informática
Universitat Politècnica de València
46022 Valencia, Spain
{rpallardo,jesparza,rgarcia,hendrik,fmunyoz}@iti.upv.es

Abstract. In the era of cloud computing and huge information systems, distributed applications should manage dynamic workloads; i.e., the amount of client requests per time unit may vary frequently and servers should rapidly adapt their computing efforts to those workloads. Cloud systems provide a solid basis for this kind of applications but most of the traditional relational database systems are unprepared to scale up with this kind of distributed systems. This paper surveys different techniques being used in modern SQL, NoSQL and NewSQL systems in order to increase the scalability and adaptability in the management of persistent data.

Keywords: Cloud Computing, Scalability, Data Management, High Availability, Distributed System, NoSQL.

1 Introduction

Scalability, *pay-per-use utility model* and *virtualisation* are the three key characteristics of the *cloud computing* paradigm. Many modern distributed applications are service-oriented and can be easily deployed in a cloud infrastructure. One of the main difficulties for achieving scalability in a cloud-based information system can be found in the management of persistent data, since data have traditionally been stored in secondary memory and replicated in order to overcome failures. As a result, this management necessarily implies noticeable delays.

In order to increase scalability while maintaining persistent data, some systems simplify or even eliminate transactions [1,2,3,4,5] in order to reduce synchronisation needs, avoiding full ACID guarantees and using simple operations that only update a single item. In contrast, other papers [6] and systems [7,8,9] still consider that regular transactions are recommendable and they need to be supported, contradicting the former. The first kind of systems is known as *NoSQL* systems, whilst systems that maintain the relational model with transactions and ACID guarantees that improve scalability applying a set of new mechanisms, are becoming to be known as *NewSQL* systems.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 208–217, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Besides, other papers [10] argue that the key elements could be the use of *idempotent actions* and *asynchronous propagation*, generating thus a *relaxed consistency* that should be assumed by application programmers. Therefore, no complete agreement on the set of mechanisms to be used in order to obtain a scalable system exists nowadays.

The aim of this paper is to provide a general description of the techniques and mechanisms that the application designer of a modern distributed information system must follow in order to guarantee an acceptable level of adaptability and scalability in data management. We also outline a general comparison of some of these systems.

The rest of the paper is structured as follows. Section 2 presents the general mechanisms to achieve scalability. Later, Section 3 summarises the main characteristics of some relevant scalable systems and, finally, Section 4 concludes the paper. An extended version of this paper can be found in [11].

2 Scalability Mechanisms

Intuitively, a distributed system is *scalable* if it is able to increase its computing power in order to deal with increasing workloads. To this end, two different approaches exist: vertical scalability and horizontal scalability. In the *vertical* case, the computing capacity of each node should be increased; for instance, expanding its memory or upgrading its CPU. In the *horizontal* case, the system is extended including additional nodes to it and, in an ideal world, when the set of nodes that compose the system was extended, a linear scalability would be obtained. Nowadays the *horizontal* approach is the most used and, from now on, we are going to assume this type of scalability in this paper.

Some characteristics of scalable data systems have been suggested in different papers [1,2] and they have been widely followed in the design of modern data scalable systems. The aim of this section is recall those recommendations and to note that none of these characteristics (or as we call, mechanisms) comes for free, since there are some trade-offs between them.

The mechanisms to be considered are:

M1 *Replication.* Data must be replicated. This allows that different server nodes hold copies of the data and each of such servers could be placed close to a given set of clients, minimising the time needed for propagating the requests and replies exchanged by clients and servers. If replication is used, *failure transparency* will be provided so, when any of the components failed, the user would not be able to perceive such failure.

Assuming a ROWAA (read one, write all available) model, replication is able to ensure linear scalability when read-only requests are considered. Note that in that case the workload can be divided among all data replicas, and no interaction is needed among them.

Unfortunately, updates always need some interactions among servers. Update propagations introduce non-negligible delays and they might prevent the system from scaling. As a result, different complementary rules should be considered for minimising those delays.

M2 *Data partitioning*. Since the set of data being managed in a modern cloud system could easily reach Petabyte sizes [3,12], it is impossible to maintain an entire copy of all these data in each of the server nodes. As a result, some kind of *partial replication* should be adopted; i.e., only some subset of the data is stored in each server.

However, *partial replication* introduces the risk of requiring multiple nodes for serving a single read-only query, since the set of items to be accessed might not be allocated to a single server. So, when the set of possible queries is known in advance, a refinement of the *partial replication* strategy called *database partitioning* [13] can be considered: to partition the data in disjoint subsets, assigning each data subset to a different server.

Database partitioning has been recommended in most systems maintaining large stores [12,14,15,8,4,16,17,5,1,6] which, in order to minimise service delays, recommend a *passive* (or primary-backup) replication model. In that model, conflicts among concurrent requests can be locally managed by the *primary* server and the need of coordination with other replicas is eliminated (if we only consider the steps related to conflict detection and transaction ordering).

M3 *Relaxed consistency.* In a distributed system, *consistency* usually refers to bounding the divergence among the states of multiple replicas of a given piece of memory. The strongest models require a complex coordination among replicas but provide a very comfortable view for the application programmer (almost identical to that of a single machine), while the most relaxed ones are able to admit multiple differences among replicas' states and they minimise the coordination needed by system nodes, but they are very hard to programme.

Regarding consistency, the key for guaranteeing a minimal delay when client operations should be managed by a replicated data store is to select a relaxed consistency model. Most modern data systems have adopted the *eventual consistency* [18] model. Such model requires that, in the absence of further updates, the states of all item replicas eventually converge. If we consider that previous principles have advised a partitioned store with a passive replication model, this allows us to use *lazy propagation* [19] (also known as *asynchronous replication*) of updates.

M4 *Simple operations.* If data operations are protected by transactions, data stores should provide concurrency control mechanisms in order to guarantee isolation, logs for ensuring data durability, and different levels of buffering and caching in order to maintain an acceptable performance level. All these managements demand a high computing effort and many I/O accesses, so it is recommended [1,2] to avoid such costs in a scalable system. The immediate effect of such attempt might be to eliminate transactions or to simplify them, only allowing single-item accesses in each transaction.

Single-item operations do also simplify the design of partitioned databases because operations will not need to access more than one partition and therefore, no algorithm will be needed to obtain a perfect database partitioning. Moreover, if we turn these operations into *idempotent* ones [1,10,20,2], we will achieve that their effects do not depend on how many times the operation is executed. Thus, if unreliable communication protocols were used, application semantics can be ensured with an *at-least-once* message delivery policy.

A final requirement that simplifies the design of recovery protocols for previously failed replicas consists in guaranteeing that all updating operations were *commutative* [10]. This recommendation is specially important in systems that assume an asynchronous multi-primary replication model.

M5 *Simple schemas.* Relational databases provide an SQL interface that is assumed by most programmers when they use a database. Unfortunately, a relational schema admits some operations (joins, for instance) that would be difficult to support in a distributed environment where the database has been partitioned (*Mechanism M2*) and the amount of server coordination steps needs to be minimised. Because of this, many scalable data stores [3,4,5,21] have renounced to the relational model and have adopted a simpler single-table *key-value* [5] schema.

These systems are becoming to be known as *NoSQL* systems.

M6 *Limited coordination.* In spite of needing a minimal server coordination, scalable data stores should maintain some meta-data (for instance, which are the current data partitions and which has been the assignment of primary replicas to each partition) whose availability is critical. So, meta-data is also replicated but it cannot follow the loosely consistent model described above for the regular store contents: its consistency should be strong and should be managed by a specialised mechanism that provides strong coordination between nodes. As there is no agreement on the name given to the set of nodes that manage these meta-data, we suggest the term *kernel set* for that mechanism.

Recent examples of *kernel sets* in scalable data stores are: the Chubby service [22] in Google's Bigtable clouds, Elastra's Metadata Manager and Master (MMM) [17] component, ZooKeeper [23] in Cassandra-based [21] and Yahoo!'s [4] systems, the Paxos Service component [24] in the Boxwood architecture, etc.

In order to sum up, notice that *Mechanism M1 (Partial replication)* does not admit any objection since all distributed systems require failure transparency and it compels some kind of replication. On the other hand, all remaining mechanisms (*M2 to M6*) do not perfectly match the regular deployment of common data services in a distributed system.

3 Scalable Systems

This section presents several scalable systems, describing which combinations of the mechanisms explained above are actually used in them. To this end, Section

Table 1. Characteristics of some scalable data management systems

	Systems				
Mechanisms	Key-value stores	Hyder	Megastore	SQL Azure	VoltDB / H-Store
Data partitioning (*M2*)	Hor.(+Vert.)	No	Horiz.	Horiz.	Horiz.
Consistency (*M3*)	Eventual	Strong	Multiple	Sequential	Sequential
Update prop.	Async.	Cache-upd.	Sync.	Async.	None
Simple operations (*M4*)	Yes	No	No	No	No
Concur. ctrl.	No	MVCC	MVCC	Yes	No
Isolation	No	MVCC	MVCC	Serialisable	Serial
Transactions	No	Yes	Yes	Yes	Yes
Simple schema (*M5*)	KeyValue	Log	KeyValue	No	No
Coordination (*M6*)	Minimal	Medium	Medium	Medium	Minimal
Admin. tasks	Yes	No	Yes	Yes	Yes
Transac. start	No	No	No	No	Yes
Dist. commit	No	Cache-upd	Yes	At times	No (active repl.)

3.1 groups the set of data stores that follow the *key-value* schema suggested by *Mechanism M5*, whilst Sections 3.2 to 3.5 describe some of the systems that do not follow such recommendation. Table 1 summarises these relationships.

3.1 Key-Value Stores

The term *key-value* store encompasses a large set of data storing systems, with some common attributes. Following *Stonebraker et al.* [25], as well as the information collected from different sources, *key-value* stores can be classified in three types:

- *Simple key-value stores*: Systems which store single key-value pairs, and provide very simple insert, delete and lookup operations. The different values can be retrieved by the associated keys, and that is the only way of retrieving objects. The values are typically considered as blob objects, and replicated without further analysis. It is the simplest approach and provides very efficient results. Some examples are Dynamo [5], Voldemort, Riak and Scalaris.
- *Document stores*: Systems which store documents, complex objects mainly composed of key-value pairs. The core system is still based on a key-value engine, but extra lookup facilities are provided, since objects are not considered just black boxes. In this way, the documents are indexed and can be retrieved by simple query mechanisms based on the provided key-value pairs. Some examples are SimpleDB [26], CouchDB, MongoDB and Terrastore.
- *Tabular stores*: They are also known as (wide) column-based stores. These systems store multidimensional maps indexed by a key, which somehow provides a tabular view of data, composed of rows and columns. These maps can be partitioned vertically and horizontally across nodes. Column-oriented

data stores are very well suited for storing extremely big tables without observing performance degradation, as some real implementations have proved. Some examples are Bigtable [3], HBase, Hypertable, Cassandra [21] and PNUTS [4].

These data stores typically implement all recommended scalability mechanisms cited in Section 2. As a result, they are able to reach the highest scalability levels. Their main limitation is a relaxed consistency that might prevent some applications from using those systems. However, other modules in a cloud system are still able to mask such problems, enforcing stricter consistency guarantees.

3.2 Megastore

Google *Megastore* [15] is a layer placed on top of a *key-value* database (concretely, Bigtable [3]) with the aim of accepting regular ACID transactions with an SQL-like interface in a highly scalable system.

The *entity group* [1] abstraction is used in order to partition the database. Each *entity group* defines a partition, and each partition is synchronously replicated (i.e., with synchronous update propagation) and ensures strong consistency among its replicas. Replicas are located in different data-centres. Therefore, each entity group is able to survive regional "disasters". On the other hand, consistency between different entity groups is relaxed and transactions that update multiple entity groups require a distributed commit protocol. So, inter-entity-group transactions are penalised and they will be used scarcely.

Transactions use multi-versioned concurrency control. They admit three different levels of consistency: *current* (i.e., strong), *snapshot* and *inconsistent* (i.e., relaxed) [15]. So, this system ensures strong consistency thanks to its synchronous update propagation but it is also able to by-pass pending update receptions when the user application may deal with a relaxed consistency, thus improving performance.

Note that Megastore is bound to the schema provided by Bigtable. So, it is compelled to use a simple schema that is not appropriate for relational data management. So, some "denormalisation" rules are needed for translating regular SQL database schemas, implementing them in Megastore/Bigtable. To this end, Megastore DDL includes a "STORING" clause and allows the definition of both "repeated" and "inline" indexes.

Regarding coordination, as we have previously seen, distributed commit protocols are needed at times in Megastore. Besides this, every regular Megastore transaction uses a simplified variant of the Paxos [27] protocol for managing update propagation. Therefore, coordination needs may be strong in some cases.

3.3 Hyder

Hyder [28] is a *data sharing* system that enhances scalability sharing a single data store between multiple servers. Its architecture still assumes relational databases and horizontal partitioning. It uses a shared set of networked flash stores (notice that the use of a networked data store could introduce a bottleneck for achieving extreme scalability levels). Furthermore, inter-server communication is kept to a minimum in this architecture since each server is able to manage its transactions using only local concurrency control mechanisms based on multi-versioning. Besides this, distributed commit is unneeded since each transaction does only use a single server.

Another important aim of this system is the usage of NAND flash memory in order to store its persistent data, since its performance and costs are improving at a fast pace. However, flash memory imposes severe restrictions on the way read/write operations and storage is administered. So, the usage of log-based file systems and a redesign of several components of the database management system [28,29] are needed. As a result, the design of Hyder is based on a log-structured store directly mapped to a shared flash storage that acts as the database: write operations will be always kept as appended records at the end of the database log while read operations (queries) are managed using local caches in each server node.

Although *Coordination* is unneeded (distributed commit protocols are avoided), since each transaction can be served by a single node (no remote subtransaction is needed), once update transactions are completed, messages should be sent to remote nodes in order to update their caches.

In order to sum up, Hyder is able to ensure strong consistency and a good scalability level without respecting all suggestions given by the mechanisms described in our paper. Nevertheless, data are actually replicated at the file system level and transactions are admitted with a SQL-like interface that might be provided on top of Hyder (although details are not discussed in [28]), making possible a regular ACID functionality.

3.4 SQL Azure

As it has been shown in the last subsections, ACID transactions may be needed in highly scalable systems, and different solutions to this lack have been proposed in Megastore and Hyder. However, none of those solutions have left the *simple schema* recommended in *Mechanism M5*. Because of this, all those systems still provide a good scalability level but they are unable to manage a fully compliant SQL interface, and such functionality is required by a large set of companies that plan to migrate their IT services to the cloud with a minimal programming effort. Note that in this case, most of those companies are more interested in database outsourcing (due to the saving in system administration tasks and in hardware renewal costs) than in extreme scalability. *Microsoft SQL Azure* [8,30] fills this void.

Obviously, in this kind of systems *Mechanism M5* should be forgotten and a regular relational schema must be adopted instead. This implies a small sacrifice in scalability, since relational databases need specialised management regarding buffering, query optimisation, concurrency control, etc. in order to guarantee all ACID properties. However, although *Mechanism M4* should also be dropped, since the aim is to fully support ACID SQL transactions, all other mechanisms (i.e., *M1*, *M2*, *M3* and *M6*) discussed in our paper are still followed.

To this end, databases are passively replicated and horizontally (or, at least, with a table-level granularity) partitioned and asynchronous propagation of updates to the secondary replicas are allowed, reducing the perceived transaction completion time.

A distributed commit protocol might be needed, but only in case of transactions that access items placed in multiple database partitions. This will not be the regular case in this system.

3.5 VoltDB

VoltDB [31] and its previous prototype *H-Store* have similar aims to those of SQL Azure: to maintain a relational schema and ACID SQL transactions in a scalable data store. However, there is an important difference between both proposals: SQL Azure is the data store for a public cloud provider company, whilst VoltDB is mainly designed for a private cluster.

VoltDB/H-Store follows the design recommendations given in [6] for improving the scalability of relational DBMSs: horizontal partitioning, main-memory storage, no resource control, no multi-threading, shared-nothing architecture (complemented with partitioning) and high-availability (replication).

Its database partitioning schema is designed by the developer, who decides if a table must be horizontally partitioned by a certain column (*partition column*) or if, in contrast, a table must be replicated. With this, distributed commit protocols are not needed and the required distributed coordination efforts are minimal: they only consist in monitoring the state of each node, reacting to node joins and failures. Furthermore, since most modern applications are not interactive, their transactions can be implemented as stored procedures and executed sequentially from begin to end without any pause, discarding multi-threading and local concurrency control mechanisms.

In summary, VoltDB is designed for systems that need SQL and transactional support with full ACID guarantees and whose data fits in main memory. Unfortunately, this system is not allowed to grow elastically: if new nodes have to be added while the system is running, it is halted in order to be reconfigured. Notice that since concurrency is avoided in this architecture, a workload dominated by long transactions would be difficult to manage in this system.

4 Conclusions

The current paper resumes which are the essential mechanisms in order to improve the scalability of persistent data storing services. It briefly describes such mechanisms and provides some pointers to systems and research papers that have adopted them or have proposed other complementary techniques.

We have seen that most key-value stores are able to directly implement all these recommended mechanisms, but provide a data consistency model that might be too relaxed. Additionally, simple operations and schemas seem to prohibit the usage of ACID transactions and relational stores. So, other systems were designed to by-pass some of these mechanisms but still achieving comparable levels of scalability. We have also presented a short summary of each of these systems, providing references that would be useful to deepen in the knowledge of this field.

Acknowledgements. This work has been supported by EU FEDER and Spanish MICINN under research grants TIN2009-14460-C03-01 and TIN2010-17193.

References

1. Helland, P.: Life beyond distributed transactions: an apostate's opinion. In: 3rd Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA, pp. 132–141 (2007)
2. Finkelstein, S., Jacobs, D., Brendle, R.: Principles for inconsistency. In: 4th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA (2009)
3. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.: Bigtable: A distributed storage system for structured data. In: 7th Symp. on Operat. Syst. Design and Implem. (OSDI), pp. 205–218. USENIX Assoc., Seattle (2006)
4. Cooper, B.F., Baldeschwieler, E., Fonseca, R., Kistler, J.J., Narayan, P.P.S., Neerdaels, C., Negrin, T., Ramakrishnan, R., Silberstein, A., Srivastava, U., Stata, R.: Building a cloud for Yahoo! IEEE Data Eng. Bull. 32, 36–43 (2009)
5. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels, W.: Dynamo: Amazon's highly available key-value store. In: 21st ACM Symp. on Operat. Syst. Princ. (SOSP), Stevenson, Washington, USA, pp. 205–220 (2007)
6. Stonebraker, M., Madden, S., Abadi, D.J., Harizopoulos, S., Hachem, N., Helland, P.: The end of an architectural era (it's time for a complete rewrite). In: 33rd Intnl. Conf. on Very Large Data Bases (VLDB), pp. 1150–1160. ACM Press, Vienna (2007)
7. Lomet, D.B., Fekete, A., Weikum, G., Zwilling, M.J.: Unbundling transaction services in the cloud. In: 4th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA (2009)
8. Campbell, D.G., Kakivaya, G., Ellis, N.: Extreme scale with full SQL language support in Microsoft SQL Azure. In: Intnl. Conf. on Mngmnt. of Data (SIGMOD), pp. 1021–1024. ACM, New York (2010)
9. Levandoski, J.J., Lomet, D., Mokbel, M.F., Zhao, K.K.: Deuteronomy: Transaction support for cloud data. In: 5th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA, pp. 123–133 (2011)
10. Helland, P., Campbell, D.: Building on quicksand. In: 4th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA (2009)
11. Muñoz-Escoí, F.D., García-Escrivá, J.R., Pallardó-Lozoya, M.R., Esparza-Peidro, J.: Managing scalable persistent data. Technical Report ITI-SIDI-2011/003, Instituto Tecnológico de Informática, Universitat Politècnica de València, Spain (2011)
12. Agrawal, D., El Abbadi, A., Antony, S., Das, S.: Data management challenges in cloud computing infrastructures. In: 6th Intnl. Wshop. on Databases in Networked Information Systems (DNIS), Aizu-Wakamatsu, Japan, pp. 1–10 (2010)

13. Stonebraker, M.: The case for shared nothing. IEEE Database Eng. Bull. 9, 4–9 (1986)
14. Alonso, G., Kossmann, D., Roscoe, T.: SwissBox: An architecture for data processing appliances. In: 5th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA, pp. 32–37 (2011)
15. Baker, J., Bond, C., Corbett, J.C., Furman, J.J., Khorlin, A., Larson, J., Léon, J.M., Li, Y., Lloyd, A., Yushprakh, V.: Megastore: Providing scalable, highly available storage for interactive services. In: 5th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA, pp. 223–234 (2011)
16. Curino, C., Jones, E.P.C., Popa, R.A., Malviya, N., Wu, E., Madden, S., Balakrishnan, H., Zeldovich, N.: Relational cloud: A database-as-a-service for the cloud. In: 5th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA, pp. 235–240 (2011)
17. Das, S., Agrawal, D., El Abbadi, A.: ElasTraS: An elastic transactional data store in the cloud. CoRR abs/1008.3751 (2010)
18. Vogels, W.: Eventually consistent. Commun. ACM 52, 40–44 (2009)
19. Breitbart, Y., Korth, H.F.: Replication and consistency: being lazy helps sometimes. In: 16th ACM Symp. on Princ. of Database Syst., PODS 1997, pp. 173–184. ACM, New York (1997)
20. Brantner, M., Florescu, D., Graf, D.A., Kossmann, D., Kraska, T.: Building a database on S3. In: Intnl. Conf. on Mngmnt. of Data (SIGMOD), pp. 251–264. ACM Press, Vancouver (2008)
21. Lakshman, A., Malik, P.: Cassandra: a decentralized structured storage system. Operating Systems Review 44, 35–40 (2010)
22. Burrows, M.: The Chubby lock service for loosely-coupled distributed systems. In: 7th Symp. on Operat. Syst. Design and Implem. (OSDI), pp. 335–350. USENIX Assoc., Seattle (2006)
23. Junqueira, F.P., Reed, B.: The life and times of a ZooKeeper. In: 28th Annual ACM Symp. on Princ. of Distrib. Comp. (PODC), p. 4. ACM Press, Calgary (2009)
24. MacCormick, J., Murphy, N., Najork, M., Thekkath, C.A., Zhou, L.: Boxwood: Abstractions as the foundation for storage infrastructure. In: 6th Simp. on Operat. Syst. Design and Impl. (OSDI), pp. 105–120. USENIX Assoc., San Francisco (2004)
25. Stonebraker, M., Cattell, R.: Ten rules for scalable performance in "simple operation" datastores. Commun. ACM 54, 72–80 (2011)
26. Amazon Web Services LLC: Amazon SimpleDB (2011), http://aws.amazon.com/simpledb/
27. Lamport, L.: The part-time parliament. ACM Trans. Comput. Syst. 16, 133–169 (1998)
28. Bernstein, P.A., Reid, C.W., Das, S.: Hyder - a transactional record manager for shared flash. In: 5th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA, pp. 9–20 (2011)
29. Bonnet, P., Bouganim, L.: Flash device support for database management. In: 5th Biennial Conf. on Innov. Data Syst. Research (CIDR), Asilomar, CA, USA, pp. 1–8 (2011)
30. Microsoft Corp.: Windows Azure: Microsoft's cloud services platform (2011), http://www.microsoft.com/windowsazure/
31. VoltDB, Inc.: VoltDB technical overview: Next generation open-source SQL database with ACID for fast-scaling OLTP applications (2010), Downloadable from: http://voltdb.com/_pdf/VoltDBTechnicalOverviewWhitePaper.pdf

Communications in Global Software Development: An Empirical Study Using GTK+ OSS Repository

Liguo Yu[1], Srini Ramaswamy[2], Alok Mishra[3], and Deepti Mishra[3]

[1] Computer Science and Informatics, Indiana University South Bend, South Bend, IN, USA
ligyu@iusb.edu
[2] Industrial Software Systems, ABB Corporate Research Center, Bangalore, India
srini@ieee.org
[3] Department of Computer & Software Engineering, Atilim University, Incek, Ankara, Turkey
{alok,deepti}@atilim.edu.tr

Abstract. Effective communication is an important issue for global software development. Due to geographical limitations and travel challenges, face-to-face meetings are expensive to schedule and run. Web-based communication methods are thus the primary means of communication in global software development efforts. In general, two types of web-based communication mechanisms exist: synchronous and asynchronous communications; each serves a unique role. In this paper, we present an empirical study of the communication mechanisms in GNOME GTK+, a small-sized open-source distributed software project, in which Internet Relay Chat (IRC) and Mailing Lists are used as synchronous and asynchronous communication methods, respectively. The objective of this study is to identify how real time and asynchronous communication methods could be used and balanced across global software development projects.

Keywords: Global software development, communications, empirical study.

1 Introduction

Global software development (GSD) is becoming an increasingly major trend for software systems development. Major software development companies have established their oversea research and development sites for years, such as Microsoft [1], IBM [2], and Oracle [3] and they are beginning to reap the benefits. For examples, products, such as Windows 7 are the result of globalized software development efforts [4]. Due to the many benefits of globalization, from the integration of multiple ethnic / market perspectives driven idea generation to development cost structuring, middle and small-sized software companies are beginning to establish worldwide development campuses / partners. Thus globalization has become an overwhelming phenomenon in the software industry and is rapidly defining the nature of software development in this first decade of the 21st century [5]. GSD, by its very nature, features distributed teams: where software developers are dispersed and spread around the world [6] [7]. Due to the geographical limitations, face-to-face meetings are expensive, and sometimes inefficient, if not supported by technology for the intervening

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 218–227, 2011.

© Springer-Verlag Berlin Heidelberg 2011

time periods between such face-to-face meetings, due to workforce availability and mobility issues. Computer network (more specifically, web) based communications thus play a vital role in requirement clarification, bug reporting, issue resolution, and commit notifications.

There are many web-based communication methods, such as mailing lists, web conferences, instant messages, and wiki. Web-based communication methods can in general, be divided into two main categories depending upon the reaction time intervals between the participants: synchronous and asynchronous communication mechanisms. Synchronous, or real-time, communication includes web conferences, instant messages, tweets, chats, etc., where developers need to remain connected during the communication process; wherein a message is sent, read instantly and possibly acknowledged with no apparent delay. Asynchronous communications on the other hand, includes mailing lists, message boards, etc., where developers do not need to remain connected concurrently during the communication process; whereby a message is sent, read, and replied to, with certain acceptable time delays.

The benefits of real time communication include fast issue resolution and fast requirement clarification. The disadvantage of real time communication include: (i) difficulty in scheduling, especially for developers located in different countries across varying time zones; and, (ii) it is suited for short messages only; long messages are not very feasible, because of the communication constraints imposed by the medium. Benefits of asynchronous communication include: (i) less urgency for responses; the developer can consider the issue presented more carefully before replying; (ii) ability to explain complicated issues with more clarity and better articulated justification and reasoning. The disadvantage of asynchronous communication is that it might take an arbitrarily longer period of time to receive a response, or a 'no response' if a developer chooses to ignore the requesting message. Nevertheless, for GSD projects, both of these communication mechanisms have to be considered. And, striking the right balance between using these two different communication mechanisms might not be a trivial undertaking. It should be based upon properties of a specific project, preferences of the developers, differences in time zones, and other specific project / organizational related issues.

In this paper, we present a case study on GNOME GTK+ project and draw some conclusions that may be useful for such projects in general. We investigate both real-time communication (Internet Relay Chat) and asynchronous communication (Mailing List) method used through the project. The principle objective of this study is to find out how real-time and asynchronous communications are used in GSD, especially for small-sized projects. In specific, we are studying the quantitative aspects of such communications from a content-agnostic perspective, i.e. the intent is not examine the content of the messages for why each method is used; so that we can derive some general observations. It has been our observation and experience that suitable GSD projects with a larger scope can often be appropriately factored, whereby these partitions can be pseudo-independently developed by smaller groups geographically distributed. It is to be noted that this study is inspired, in part, by the work in [8].

The rest of this paper is organized as follows. Section 2 describes GNOME GTK+ project and its archive of synchronous (Internet Relay Chat—IRC) communication mechanism and archive of asynchronous (Email List) communication mechanism.

Section 3 describes the data mining process used on these archives. Section 4 presents our results and the analysis of the observations. Our conclusions on the case study are summarized in Section 5.

2 Communications in GNOME GTK+ Project

GNOME GTK+ is a small-sized open-source toolkit with rich features, cross-platform compatibility and an easy to use API for creating graphical user interfaces. GTK+ is originally written in C, and has bindings to many other popular programming languages such as C++, Python and C# [9].

The stable version of GTK+ (v1.0) was released in April 1998. Until now (September 2010), 13 versions have been released. The latest version (v2.20) was released in April 2010 [10]. Currently, there are nine core maintainers in GTK+ project [11]. There have been hundreds (near one thousand) of developers who have participated in the development of GTK+. Two major communication mechanisms have been used by GTK+ developers: developer mailing list and Internet Relay Chat— IRC.

GTK+ developer mailing list (gtk-devel-list@gnome.org) is an asynchronous communication method. It is used by developers to discuss the design and implementation issues of core GTK+ libraries. GTK+ application development and general GTK+ questions are handled by different mailing lists. Bug reports are entered and handled through Bugzilla.

As regularly as possible, GTK+ team meetings take place in the Internet Relay Chat (IRC) channel on irc.gnome.org (#gtk-devel). It is a synchronous, real-time communication method. Everyone is welcome to join the meeting. However, the ground rule is that the channel is to be used only for GTK+ team meetings, and not for general questions about GTK+.

Therefore, both GTK+ developer mailing list (gtk-devel-list@gnome.org) and the IRC channel (#gtk-devel) are used to discuss the core development issues of GTK+. The developer mailing archive contains data back to 1997 and IRC meetings are recorded and the data is available dating back to 2004. Figure 1 illustrates the mailing list structure: an email thread and a detailed message, where the highlighted element indicates the time stamp and the time zone info. Figure 2 illustrates the IRC meeting log structure with the message sender email ID highlighted.

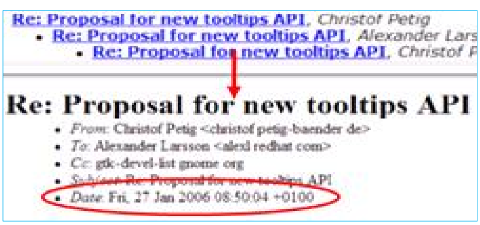


Fig. 1. An email thread and a detailed message

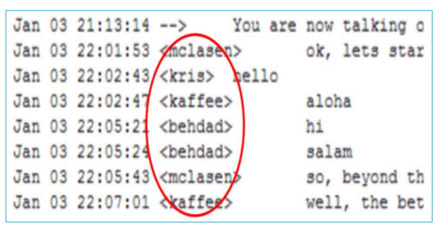


Fig. 2. Part of the IRC meeting log

3 Data Mining Process

Perl programs were written to mine the developer mailing list [12] and the IRC record [13] [14]. For each email message and IRC meeting the following information was retrieved from the data mining process and saved to a text file for further analysis.

Email Message:	IRC Message:
• ID (every message is assigned a unique id) • Parent message ID (For new message, parent message ID is assigned value 0; for replying message, parent message ID is the ID of the original message) • Author • Subject • Date and time • Time zone	• Date • Participants • Posters of each message

In the mailing list, the email message time stamps, recorded as local time, were converted to GMT (UTC) time, to enable analysis. Similarly, to enable a comparative study, since the IRC meeting records were available only after 2004, we correspondingly analyzed the mailing list record back to the same year 2004.

4 Analysis and Results

4.1 The Correlations between Synchronous and Asynchronous Communication

From February 2004 to November 2009, the total number of email threads and the total number of email messages are listed in Table 1; the total number IRC meetings and the total number of messages posted in these meetings are listed in Table 2. In mailing list, an email thread contains one root message with parent ID 0, and zero or more replying messages with non-zero parent ID. For example, Figure 1 illustrates one email thread and ten email messages. In contrast, Figure 2 illustrates one IRC meeting and seven messages (posts).

In IRC communications, each meeting is planned to discuss an apriori determined number of issues. Therefore, the number of meetings is correlated with the number of issues discussed; and the number of messages posted in these meetings represents the complexity of the issues. Similarly in mailing lists, each email thread represents one issue. Therefore, the number of email threads represents the number of issues discussed in mailing lists and the number of email messages in each thread represents the complexity of each issue.

To study whether synchronous (real-time) communication (IRC) and asynchronous communication (mailing list) were correlated in discussing issues, we study (i) the correlation between the number of IRC meetings and the number of email threads in the same period; and (ii) the correlation between the number of messages posted in IRC meetings and the number email messages posted in mailing list in the same period. Spearman’s rank correlation tests are performed. The results of tests on yearly aggregated data are shown in Table 3.

Table 1. Number of Email threads and messages

Year	Number of threads	Number of messages
2004	308	783
2005	330	980
2006	374	1166
2007	372	1425
2008	316	1067
2009	313	983

Table 2. Number of IRC meetings and messages

Year	Number of meetings	Number of posts in meetings
2004	35	5132
2005	37	5524
2006	13	1258
2007	9	1945
2008	16	3088
2009	8	2005

Table 3. Spearman's test on yearly-based data

Variable			
Variable	x	Number of IRC meetings	Number of posts in IRC meetings
	y	Number of Email threads	Number of Email messages
Data sets		6	6
Correlation co.		-0.142	-0.828
Significance (*p*)		0.802	0.058

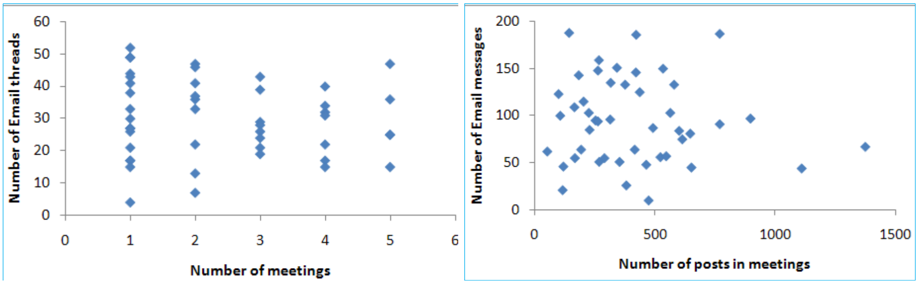
It can be seen from Table 3 that for each year, negative correlations are found (1) between the number of IRC meetings and the number of email threads; and (2) between the number of posts in IRC meetings and the number of email messages. However, none of these correlations are significant at the 0.05 level.

To further study these correlations, the data are reorganized in a monthly aggregation basis. From February 2004 to November 2009, a total of 46 month data are retrieved (months without IRC meetings are ignored). Spearman's rank correlation tests are performed to study the correlations on monthly-based data. The results are summarized in Table 4.

Again for each month, negative correlations are found (1) between the number of IRC meetings and the number of email threads; and (2) between the number of posts in IRC meeting and the number of email messages. However, these correlations are not significant at the 0.05 level. The scatter plots in Figure 3 illustrate their relationships graphically.

Table 4. Spearman's test on monthly-based data

Variable	x	Number of IRC meetings	Number of posts in IRC meetings
	y	Number of Email threads	Number of Email messages
Data sets		46	46
Correlation co.		-0.128	-0.083
Significance (*p*)		0.394	0.583



(a) (b)

Fig. 3. The scatter plots of monthly data of (a) number of IRC meetings versus number of email threads; and (b) number of posts in IRC meeting versus number of email messages

Because negative correlations are found between (1) the number of meetings and the number of email threads, and (2) the number of posts in IRC meetings and the number of email messages, it seems that real-time communication method (IRC) and asynchronous communication method (email) are used complementarily, i.e. the more usage of one method corresponds to the less use of another method. However, because none of these correlations are significant at the 0.05 level, these findings are inconclusive and they are only our speculations.

4.2 Exploring Communication Participants

Table 5 shows the number of participants in GTK+ IRC meetings and the number of participants in mailing list communications of each year. A/R is defined as the ratio of the number of asynchronous communication participants (mailing list) to the number of real-time communication participants (IRC meeting). It can be seen that this value is in the range of 3 to 7. In most years, this value is about 4, i.e. about 4 times the number of developers participated using asynchronous communication (mailing list) compared with those, who participated using real-time communication (IRC meeting).

Table 5. GTK+ project communication participants

Year	IRC meetings	Mailing list	*A/R*
2004	49	158	3.2
2005	71	184	2.6
2006	32	209	6.5
2007	55	214	3.9
2008	73	190	2.6
2009	44	190	4.3

Exploring further, we also studied the activeness of GTK+ communication participants. It has been observed that in both synchronous communication and asynchronous communication, majority of the messages are contributed by a small number of participants. In both communication mechanisms, active contributors are defined as the top active posters who have contributed to over 80% of all the messages (email message or IRC post) in each year. Table 6 shows the number of active contributors and total contributors in mailing list and Table 7 shows the number of active contributors and total contributors in IRC meetings, where A/T is defined as the percentage of total contributors who are active contributors.

Comparing the A/T percentages in IRC meetings (average 26%) and the A/T percentages in mailing list (average 33%), it can be seen that in general regular developers are more active through asynchronous communications. It also means that in real-time communication (IRC meetings), the discussions are led by active posters.

Table 6. GTK+ project mailing list participants

Year	Active	Total	*A/T*
2004	55	158	35%
2005	62	184	34%
2006	68	209	33%
2007	64	214	30%
2008	65	190	34%
2009	64	190	34%

Table 7. GTK+ project IRC meeting participants

Year	Active	Total	*A/T*
2004	9	49	18%
2005	13	71	18%
2006	8	32	25%
2007	19	55	35%
2008	17	73	23%
2009	15	44	34%

To see how often a developer use both communication methods, we studied the top 20 most active IRC meeting posters and the top 20 most active email posters in each year. The number of developers that belong to both top-20 lists is illustrated in Figure 4.

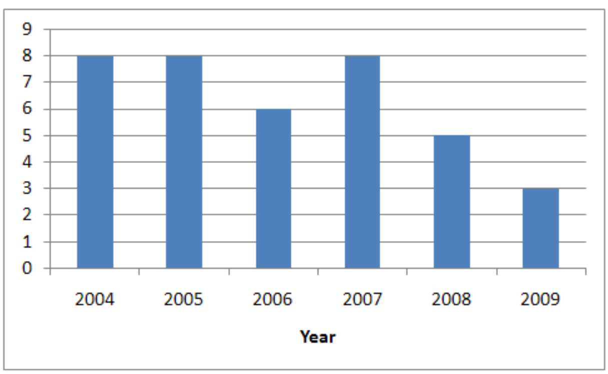


Fig. 4. The number of developers who are ranked as top 20 contributors to both synchronous and asynchronous communications

Figure 4 can be interpreted as follows: if a developer is an active communicator using one method (real-time or asynchronous), s/he is very likely to be active using a different method (asynchronous or real-time). In the GTK+ web site, there list nine core developers (Table 8). To see if these core project members are actively using both communication methods (real-time and asynchronous), we extracted the participants who have been on both of the two top-20 lists for at least one year (2004-2009) and their names are shown in Table 9.

Table 8. Current core members of GTK+ project

Name	Affiliation
Tim Janik	Lanedo GmbH
Matthias Clasen	Red Hat
Behdad Esfahbod	Red Hat
Federico Mena Quintero	Novell
Alexander Larsson	Red Hat
Tor Lillqvist	Novell
Kristian Rietveld	Lanedo GmbH
Michael Natterer	Lanedo GmbH
Emmanuele Bassi	Intel

Table 9. The most active participants and number of years on both top-20 lists

Name	No. of years	Name	No. of years
Matthias Clasen	6	Tristan Van Berkom	2
Alexander Larsson	5	David Zeuthen	1
Tim Janik	5	James M. Cape	1
Emmanuele Bassi	4	Johan Dahlin	1
Owen Taylor	3	John Ehresman	1
Behdad Esfahbod	2	Jonathan Blandford	1
Federico Mena Quintero	2	Maciej Katafiasz	1
Kristian Rietveld	2	Sven Neumann	1

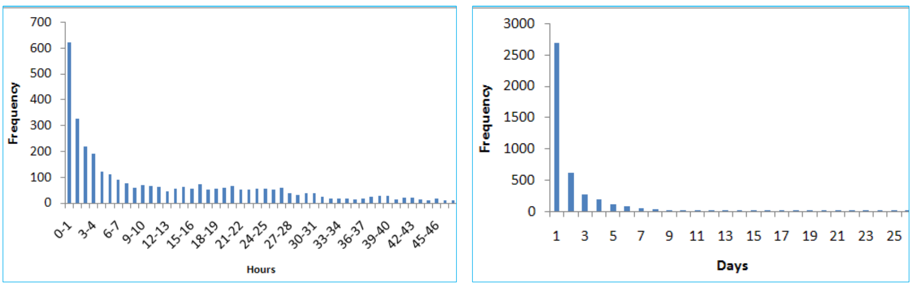
Comparing Table 8 with Table 9, it can be seen that 7 out of the 9 current core members are actively using both communication methods (real-time and asynchronous). This observation also indicates an often ignored, yet intrinsic part of many successful projects, and perhaps irrespective of the localized or distributed nature of the development activity itself, i.e., the emergence of a relative degree of stability and continuity amongst the core team participants.

4.3 Email Response Time Delay

Unlike real-time communication using IRC chat, which has no significant time delay between messages, asynchronous communication using email usually has some delays, from minutes, hours, to days. In this section, we study the time span between the original message and the replying message. The mailing list data of GTK+ project from February 2004 to November 2009 is analyzed.

Figure 5(a) shows the frequency of time delay (less than or equal to 48 hours) between the replying message and the original message. In this time delay range, most messages are replied within 24 hours. Figure 5(b) shows the frequency of time delay (all messages) between the reply message and the original message. In this time delay range, most messages are replied within 2 days.

Next, we study the effect of time zone differences on Email response delay. We assume that time spans over 48 hours are not affected by time zone difference. In other words, only if the replying message and the original has a time stamp difference less than or equal to 48 hours, time zone difference may take effect; if replying message and original has a time stamp difference greater than 48 hours, other factors have a more direct effect than the time zone. Therefore, we only studied the relation between time zone difference and response delay within 48 hours. The result is shown in Figure 6.



(a) (b)

Fig. 5. The frequency of time delay between the replying message and the original message: (a) time span is less than or equal to 48 hours; and (b) all data

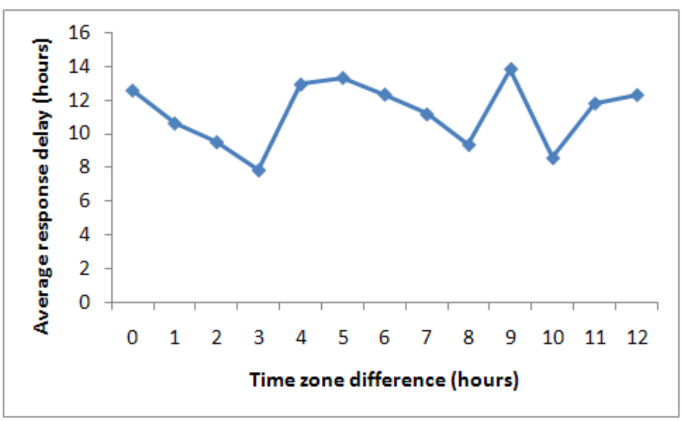


Fig. 6. The relation between time zone difference and the average email response delay within 48 hours

In Figure 6, the time zone differences are calculated as absolute values, which means the largest time zone difference is 12 hours. We can see that on average, time zone difference has no apparent effect on response delays.

5 Conclusions

In this paper, we presented an empirical study of synchronous and asynchronous communication mechanisms on GNOME GTK+, an open source distributed software development project, where developers used Internet Relay Chat (IRC) and Email lists as the communication mechanisms. We mined this communication history of GNOME GTK+ developer network. Major observations of our studies include these specific results: (1) Insignificant negative correlations exist between the amount of real time communication (IRC) and asynchronous communication (mailing list); (2) Core developers are actively using both mechanisms to communicate with their group; (3) Most emails are replied within 2 days; and, (4) Time zone difference has no apparent effect on email response delay.

The threats to validity of our study are that our research is performed on one small-sized open-source project. To further validate these results, more studies need be performed on other distributed software projects. Moreover, these observations serve as a useful starting point for some in-depth research including the addressing of other related issues such as variability in language use and cultural effects.

References

1. Microsoft Worldwide, `http://www.microsoft.com/worldwide/`
2. IBM Research, `http://www.research.ibm.com/`
3. Oracle Worldwide Site, http://www.oracle.com/global/index.html
4. Windows International Team. Engineering Windows 7, `http://blogs.msdn.com/e7/archive/2009/07/07/engineering-windows-7-for-a-global-market.aspx`
5. AeA's Board of Directors. Offshore Outsourcing in an Increasingly Competitive and Rapidly Changing World. AeA (American Electronic Association) (March 2004), `http://www.techamerica.org/content/wp-content/uploads/2009/07/aea_offshore_outsourcing.pdf`
6. Al-asmari, K.R., Yu, L.: Experiences in distributed software development with wiki. In: Proceedings of 2006 International Conference on Software Engineering Research and Practice, Las Vegas, Nevada, June 26-29, pp. 389–393 (2006)
7. Al-asmari, K.R., Batzinger, R.P., Yu, L.: Experience distributed and centralized software development in IPDNS project. In: Proceedings of 2007 International Conference on Software Engineering Research and Practice, Las Vegas, Nevada, pp. 46–51 (June 2007)
8. Shihab, E., Jiang, Z.M., Hassan, A.E.: On the use of Internet Relay Chat (IRC) meetings by developers of the GNOME GTK+ project. In: Proceedings of the 6th International Working Conference on Mining Software Repositories, Vancouver, BC, Canada, May 16-17, pp. 107–110 (2009)
9. GNOME GTK+ project, `http://www.gtk.org/`
10. Wikipedia: GTK+, `http://en.wikipedia.org/wiki/GTK%2B`
11. GTK+ development, `http://www.gtk.org/development.html`
12. GTK+ developer mailing archive, `http://mail.gnome.org/archives/gtk-devel-list/`
13. GTK+ meeting space, `http://live.gnome.org/GTK+/Meetings`
14. GTK+ team meetings, `http://www.gtk.org/plan/meeting`

Varying Levels of RFID Tag Ownership in Supply Chains

Wei Zhou[1,3], Eun Jung Yoon[2], and Selwyn Piramuthu[2,3]

[1]Information & Operations Management, ESCP Europe, Paris, France
[2]Information Systems and Operations Management, University of Florida
Gainesville, Florida 32611-7169, USA
[3]RFID European Lab, Paris, France
wzhou@escpeurope.eu, {ejyoons,selwyn}@ufl.edu

Abstract. RFID-tagged items in a supply chain experience different types of ownerships, often simultaneously, throughout their lifetime. We consider scenarios related to ownership of RFID tags in a supply chain, specifically the seamless incorporation of third party logistics provider. We develop protocol for authentication of RFID-tagged items under such conditions.

Keywords: RFID, permanent key, temporary key, distributed environment.

1 Introduction

Radio Frequency IDentification (RFID) systems are increasingly being used in distributed environments where there is a need to embed item-level information on the item of interest to improve access to necessary (item-level) information when and where they are needed regardless of network connectivity and related issues. We consider the case of third party logistics providers and their role in (temporary) ownership transfer of items in a distributed supply chain environment. Specifically, we consider the simultaneous RF access of an RFID-tagged item by the item's owner and the (temporary) owner (here, the third party logistics provider) through the use of two keys - a main key for the owner and a sub-key for the third party logistics provider.

We develop ownership transfer protocol that accomplishes this in a seamless fashion. To our knowledge, none of the existing ownership transfer protocols address this issue of multiple keys with varying levels of access to the functionalities of the RFID tag. We first present the scenario and then present the proposed protocol. We provide a brief security analysis to check for some common vulnerabilities that are present in related protocols.

This paper is organized as follows: The considered scenario, the sequence of steps for ownership transfer in this scenario, as well as the proposed protocol following these steps are presented in the next section. The last section concludes the paper with a brief discussion.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 228–235, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 The Proposed Authentication Protocol

We first present the considered scenario, followed by the sequence of events in the modeled scenario, and then the set of notations used in the proposed protocol. We then present and discuss the proposed protocol.

2.1 The Scenario

We consider a scenario in a supply chain where an RFID-tagged item changes ownership. Along the way, the RFID-tagged item also needs to have the capability to communicate with third party logistics provider(s). Whereas the former can readily be handled by generic single-tag single-owner ownership transfer protocols from extant literature, to our knowledge existing protocols do not address the scenario that interleaves the former with the latter. There exist at least four main entities/players in this system: current and next owners of the RFID tag (and, therefore, the RFID-tagged entity), the RFID tag, the third party logistics company. We consider the use of only passive tags, thereby adhering only to lightweight cryptography to develop the proposed ownership transfer/sharing protocol. Current state-of-the-art on RFID (cryptography) authentication protocols for basic passive RFID tags necessitates the use of symmetric-key cryptography. The choice of symmetric-key cryptography renders it extremely difficult, if not impossible, to perform ownership transfer without a trusted third party (TTP). Therefore, we include a TTP as an entity that participates in the proposed protocol.

The following are some of the characteristics of this system:

1. The (current and next) owner and the tag share a common secret K (the main key)
2. The third party logistics (TPL) provider i and the tag share a common secret k_i (the sub-key).
3. The owner can communicate with the tag only with the knowledge of both K (the main key) and k_i (the sub-key).
4. The TPL provider, on the other hand, can communicate with the tag using just the sub-key.
5. The Reader has 'super user' power and can modify the settings including the sub-key in the tag, whereas the TPL provider is not allowed to change any of the keys (i.e., the main and/or the sub keys).

2.2 The Event Sequence

We assume that ownership transfer occurs when an item is transported from one physical location to another. The ownership transfer process itself can occur at the origin or the destination locations or even somewhere in-between. Regardless, there are two main dynamic at play here: ownership transfer of the item from current to new owner and the granting of communication privilege/permission (i.e., shared secret sub-key) to the TPL provider so that the TPL provider can communicate directly with the tag. We assume the presence of a TTP when ownership transfer occurs as well as when the TPL provider obtains and surrenders the sub-key. Since the TPL provider obtains the sub-key at the origin of this trip-segment, we assume the presence of a TTP at the origin point. Similarly, since the TPL provider surrenders the sub-key at the destination of this trip-segment, we assume the presence of a TTP at the destination point.

There are a few variations to this scenario depending on where (i.e., origin, destination, somewhere in-between) the primary (i.e., between current and next owner) ownership transfer occurs. Regardless, given that the two owners need not simultaneously be at the same physical location, ownership transfer can be accomplished by the two TTPs (one at the origin and the other at the destination points) independent of when it occurs since all that is necessary are the physical proximity of the two (current and next) owners to either or both of the two TTPs and the existence of a (secure or unsecure) communication channel between the two TTPs. We also assume the presence of several (at least two as discussed above) TTPs and that these TTPs are all identical and are able to communicate among one another to pass necessary messages through secure channels. I.e., for our purposes, even though we consider two 'separate' TTPs, they both can be considered a single entity.

The following is the sequence of events that occur in the modeled scenario:

1. The current owner possesses (or, obtains from the TTP) the main key (i.e., K) to the item of interest.
2. The tag, current owner, and the (if any) TPL provider obtain sub-key for the item at the origin location.
3. When appropriate, the item is transported from origin to destination location.
4. The new owner obtains the main key from TTP.
5. The new owner, TPL provider, and tag obtain the updated sub-key from the TTP.

We enforce the need for any owner to have knowledge of both main- and sub-keys for communicating with the tag. The composite key thus necessary can be represented as $K \oplus k_i$, where k_i is the item's sub-key during the i^{th} period that may or may not involve a TPL provider. When the two (i.e., current and next) owners are physically at different locations, the TPL provider is assumed to transport the item between the two owners. We define a period as one when either ownership transfer occurs or the item is transported to another physical location, or both. The beginning of a new period triggers item's sub-key update.

2.3 Notations

The following notations are used throughout the paper:

- p, l, t, o, o': m-bit nonce
- K, K': tag's shared (with current and next owner) secret primary (main) key

- k_{i-1}, k_i: tag's shared (with owner, TTP) secret temporary (sub) key during the $(i-1)^{th}$ and i^{th} period respectively
- s, s': shared secret key between TTP & current and next owner respectively
- k_T: shared secret key between tag and TTP
- k_L: shared secret key between TPL provider and TTP
- f_k: keyed (with key k) encryption function

2.4 The Protocol

The proposed protocol (Figure 1) can be used for (1) generation and transfer of the main key whenever ownership transfer occurs and (2) generation and transfer of the sub-key whenever either ownership transfer or the use of a new TPL provider is involved. I.e., the main- and sub-key are updated whenever ownership transfer occurs and only the sub-key is updated whenever the RFID-tagged item is transported by a TPL provider. The main key is associated with the owner whereas the sub-key is associated with the TPL provider. The messages between any two entities as presented in Figure 1 occur through over-the-air channel that is not assumed to be secure. The entities participating in the protocol are assumed to be in close proximity to one another - at the least, any pair of communicating entities are expected to be in close physical proximity to each other.

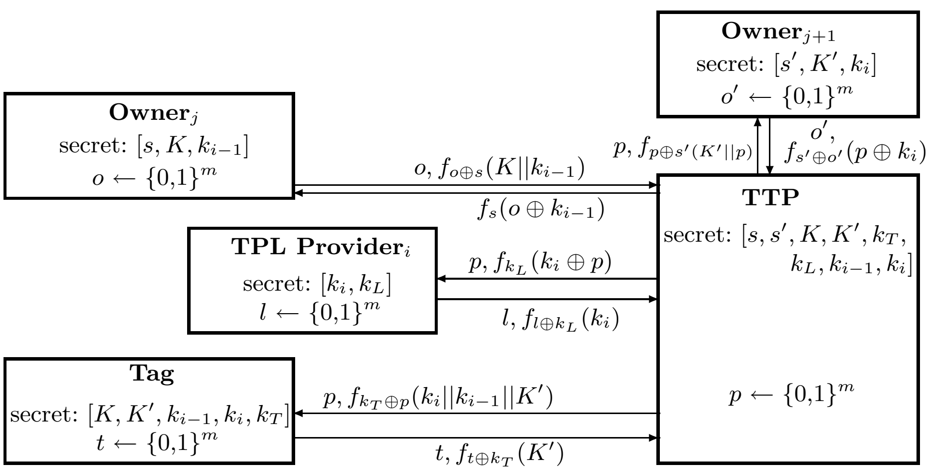


Fig. 1. The proposed ownership transfer protocol

The protocol includes several shared secret keys between pairs of entities. The shared secret keys between the current and next owners and TTP are respectively s and s'. The shared secret sub-keys between the current and next owners and TTP are respectively k_{i-1} and k_i. This sub-key is also shared with the current (if any) TPL provider. The shared main keys between the current and next owners and TTP are respectively K and K'. The shared secret key between the TPL provider and the TTP is k_L. Finally, the shared secret key between the tag and the TTP is k_T. Each of the entities generate fresh nonce when initiating a 'loop'. The set of nonce include o, o', l, t, and p respectively for the current owner, next owner, TPL provider, tag, and TTP.

The messages in the protocol form four loops: the first loop between current owner and TTP is used by the current owner to inform the TTP of the change in status of an RFID-tagged item - i.e., this tagged item either changes ownership or goes through a TPL provider or both. The second loop between the TTP and the next owner is for the TTP to generate a new main key and inform the new owner of the same. However, when the owner remains the same but there is a TPL provider involved, the TTP informs the new owner of the same and the new owner generates a new sub-key for the tag. The third and fourth loops are for the TTP to inform the TPL provider and tag respectively of the new sub-key for the tag.

The protocol begins when the current owner generates a fresh nonce (o) and encrypts its message to the TTP using its shared key and sends the message to the TTP $(o, f_{o\oplus s}(K||k_{i-1}))$. When the intention is to change just the sub-key (i.e., no ownership transfer is involved), the sub-key field is replaced with null value (i.e., $k_{i-1} \leftarrow \phi$. In response, the TTP instantiates the second loop in the protocol - the one between the TTP and the new owner (i.e., Owner$_{j+1}$). The first loop is closed at the end of the protocol. Owner$_j$ waits for a pre-determined amount of time to receive acknowledgement from the TTP. When this does not happen, Owner$_j$ generates a fresh nonce (say, o') and re-starts the process. This is repeated until successful completion of the protocol.

The TTP then updates the main key (K to K'). Depending on k_{i-1} (i.e., its value being null or otherwise), the TTP modifies its message $(p, f_{p\oplus s'(K'||p)})$ sent to Owner$_{j+1}$. Here, when the situation dictates changing only the main key (K to K'), the p value concatenated with K' is set to null. Upon receipt of this message, Owner$_{j+1}$ retrieves the new main key (i.e., K') and generates a new sub-key (i.e., k_i) when necessary. The message from Owner$_{j+1}$ (i.e., $o', f_{s'\oplus o'}(p \oplus k_i)$) is modified accordingly - i.e., when Owner$_{j+1}$ does not generate a new sub-key (k_i), the value of k_i in the encrypted message is set to zero. Once it sends its message to Owner$_{j+1}$, the TTP waits for a pre-determined amount of time to hear back. Otherwise, it generates a fresh nonce (say, p') and re-sends its message to Owner$_{j+1}$. Once it receives the acknowledgement, the TTP initiates the third loop between the tag and itself.

The third loop is initiated by the TTP by generating a fresh nonce (say, p'') and sending the message $(p, f_{k_T\oplus p}(k_i||k_{i-1}||K'))$to the tag. Although the protocol (and, therefore, this loop) is instantiated only when there's a change in key(s), this message is modified depending on what is changed (i.e., the main key and possibly the sub-key). When the sub-key is unchanged in this round of the protocol, the k_i value is set to zero. Upon reception of this message from the TTP, the tag retrieves the new main key (i.e., K') and sends an acknowledgement to the TTP with $t, f_{t\oplus k_T}(K')$. The TTP times this loop and re-initiates the loop with a fresh nonce if it does not receive a valid acknowledgement from the tag within a pre-determined amount of time.

The last (optional) loop is between the TTP and the TPL provider and this loop is used to transfer the tag's sub-key to the TPL provider. The TTP does not initiate this loop if the sub-key is not changed during this round of the protocol. Once the TPL provider successfully receives the message (i.e., $p, f_{k_L}(k_i \oplus p)$) from the TTP and retrieves the tag's sub-key (i.e., k_i), it generates a fresh nonce (l) and sends an acknowledgement message ($l, f_{l \oplus k_L}(k_i)$) to the TTP. As with the other loops in this protocol, this loop is also timed by the TTP. It repeats this loop with a fresh nonce if it does not receive the acknowledgement from the TPL provider within a pre-determined amount of time. Once this (optional) loop is completed, the TTP completes the first loop by sending an acknowledgement message (i.e., $f_s(o \oplus k_{i-1})$) to Owner$_j$. This loop is timed by Owner$_j$, which re-starts this protocol with a fresh nonce if it fails to receive TTP's acknowledgement within a pre-determined amount of time. This process continues with fresh nonce until the protocol comes to successful completion.

Every loop in the protocol is timed to ensure that all initiating messages reach their destination and the acknowledgement messages reach their destination.

2.5 Security Analysis

We briefly consider a few security violations that can arise in the presented context and evaluate the proposed protocol.

- *Denial of Service (DoS) / desynchronization attack*: Denial of Service (DoS) attacks can arise due to several reasons including desynchronization between tag and reader and can lead to disconnect between tag and reader. When this occurs, the tag (or, reader) will not be able to communicate with the reader (or, tag). In the proposed protocol, there is very little chance for desynchronization to occur since when the shared keys are updated, the entity updating the key (e.g., TTP) ensures that all parties involved (e.g., Owner, TPL provider, tag) are aware of this update through timed message loops.
- *Forward Security*: Forward security is necessary to maintain the integrity of the system and is especially critical in systems where messages are exchanged over the air. Forward security implies that even if an adversary copies every message that has been exchanged between two entities, the adversary cannot decipher any of them even though the secret keys at a later point in time are known to the adversary. This protocol does not guarantee forward security since none of the messages are encrypted using one-way hash functions. It is possible to address this issue through the use of one-way hash functions to encrypt messages throughout the protocol. However, that could lead to increased computational load on the entities, especially the tag, which is bound by extreme resource constraints. The messages sent between the TTP, Owner$_j$, Owner$_{j+1}$, and TPL provider can all be encrypted using one-way hash functions (instead of the encryption function used) to alleviate this problem.

- *Replay attack*: Replay attack occurs when an adversary passively observes communication among entities and copies those messages for later use. At some later point in time, the adversary strategically replays some of these messages, which are accepted as valid by the receiving entity. Since most messages that could be replayed in the proposed protocol include freshly generated nonce, replaying any of these will not work (i.e., will not be validated by the receiving entity).
- *Impersonation attack*: Impersonation attack occurs when an adversary is able to completely impersonate an entity to all other entities that participate in the protocol. In the proposed protocol, an adversary cannot impersonate any of the entities to another due to the use of freshly generated nonce in each of the three (and the fourth optional) loops. It is also difficult for an adversary to obtain the shared secrets (i.e., s, s', k_{i-1}, k_i, K, K', k_T, k_L).

3 Discussion

We considered a scenario that necessitates the presence of a pair of keys for each RFID tags - one permanent for the owner of this item and a temporary one for each temporary 'owner' (here, third party logistics provider) of this item. We proposed an authentication protocol for this scenario that involves four entities - the current owner, the next owner, the third party logistics provider, a trusted third party (TTP), and the tagged item. All communication in this protocol occur over the air where the communication channel is not assumed to be secure. We used lightweight symmetric cryptography to develop this protocol and discussed some of its security properties.

The use of a new sub-key every time a new third party logistics provider gains access to the RFID tag ensures maintenance of a certain degree of privacy and security of the RFID-tagged item. This also prevents (both physical and RF) unauthorized access of the RFID-tagged item by anyone including adversaries. The separation of the composite key pair into main- and sub- keys facilitates maintaining and granting different security access levels/permissions to the RFID-tagged items. Communication with the RFID-tagged item can therefore be seamlessly accomplished both by the owner with the knowledge of the permanent main key and the TPL provider with the knowledge of the temporary sub key. We also assume that, to gain access to higher level privileges the owner needs access to both the main- and the sub-keys.

While it is possible, in principle, to develop secure authentication protocols, it is difficult to ensure protection from relay attacks. Relay attacks occur when adversaries simply relay messages between a honest reader and a honest tag with or without the knowledge of either party (e.g., [1], [2], [3], [5]). The difficulty with relay attacks lies in the absence of cryptographic manipulations by the adversary. The proposed protocol, therefore, is not immune to relay attacks. This is not a major drawback given that a majority, if not all, of extant authentication protocols are not immune to relay attacks.

We considered the possibility of simultaneous update of both main- and sub-keys of the tag. Whereas the main-key can be updated without updating the sub-key (i.e., when a change of tag ownership occurs), we assumed that the sub-key is not updated without related update of the main-key. The proposed protocol only considered (main- and/or sub-) key update. We did not present protocols where the owners as well as other entities communicate with the RFID-tagged item. The presented example scenario and the proposed protocol highlight the importance of the existence of a pair of keys in each tag in the considered context.

References

1. Brands, S., Chaum, D.: Distance Bounding Protocols. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994)
2. Desmedt, Y.: Major Security Problems with the 'Unforgeable' (Feige)-Fiat-Shamir Proofs of Identity and How to Overcome Them. In: Proceedings of the Securicom 1988, 6th Worldwide Congress on Computer and Communications Security and Protection, pp. 147–159 (1988)
3. Hering, J.: The BlueSniper 'rifle,' presented at 12th DEFCON, Las Vegas (2004)
4. Juels, A.: Yoking-Proofs for RFID Tags. In: Proceedings of the First International Workshop on Pervasive Computing and Communication Security (PerSec), pp. 138–143. IEEE Press (2004)
5. Kfir, Z., Wool, A.: Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems. In: Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm), pp. 47–58 (2005)
6. Piramuthu, S.: RFID Mutual Authentication Protocols. Decision Support Systems 50(2), 387–393 (2011)

Workflow Validation Framework in Distributed Engineering Environments

Wikan Danar Sunindyo, Thomas Moser, Dietmar Winkler, Richard Mordinyi, and Stefan Biffl

Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems (CDL-Flex), Institute of Software Technology and Interactive Systems
Vienna University of Technology, Favoritenstrasse 9-11, 1040 Vienna, Austria
{wikan.sunindyo,thomas.moser,dietmar.winkler, richard.mordinyi,stefan.biffl}@tuwien.ac.at

Abstract. Automation systems, like power plants and industrial production plants, usually involve heterogeneous engineering domains, e.g., mechanical, electrical, and software engineering, that are required to work together to deliver good products and services to customers. However, the heterogeneity of workflows used in different engineering domains make it hard for project managers to integrate and validate such workflows. We propose to add the Engineering Service Bus (EngSB) notion with a framework to integrate and validate heterogeneous workflows from different engineering fields and to link connections between different types of signals. For evaluation, we perform a feasibility study on a signal change management of our industry partner in hydro power plant engineering domain. Major result shows that the framework can support workflow validation and improve the observability of heterogeneous workflows in distributed engineering environments.

Keywords: Workflow integration and validation, automation systems engineering, process mining, process observation, process improvement.

1 Introduction

Complex automation systems, like power plants or industrial production plants, involve different kinds of engineering fields, e.g., mechanical, electrical, and software engineering fields, that are required to work together to produce high quality products or services. However in distributed systems, typically each stakeholder works separately in his workplace, defining and using his own workflow to solve specific tasks. The interactions between different stakeholders are coordinated and monitored by project managers, who have the responsibility to monitor the progress of project and take action or decisions based on the current status of the project, e.g., to add more personnel or change/improve the overall engineering process.

To be able to monitor the progress of projects, project managers should have an integrated overview across different workflows used by the heterogeneous project stakeholders, such as that project managers can understand the interactions between

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 236–245, 2011.
© Springer-Verlag Berlin Heidelberg 2011

engineers and that they can validate the designed workflows compared to the actual engineering processes.

The major challenges here are (a) how to integrate heterogeneous workflows from different stakeholders to support an overview of the project progress, (b) how to validate the designed workflow model with the actual engineering processes. Current workflow validation approaches , such as the approach described by Raedts et al. [6], are only applied to homogenous and centralized systems. The application of similar approaches to heterogeneous engineering environments is hard, due to heterogeneity of notations used to represent the workflows and since the relationships between different workflows are hard to be defined automatically.

We propose to use and expand the notion of Engineering Service Bus (EngSB) [2] to integrate and validate heterogeneous workflows in distributed engineering environments. The EngSB is an approach to bridge technical and semantic gaps between engineering models and tools for quality and process improvements for Software Engineering (SE) and other engineering disciplines that interact with SE. The EngSB is based on proven concepts of the "Enterprise Service Bus" (ESB) [3] approach situated in the business IT context which are adapted to address the needs of systems engineering [1].

We improve the Process Mining[1] (ProM) tool for validating the integrated workflow model with heterogeneous event logs from different engineering fields. We evaluate our proposed approach by implementing this workflow integration and validation methods for an industrial use case in the hydro power plant engineering domain, to show the feasibility for distributed information systems. We measure the efforts required by project managers to integrate and validate the heterogeneous workflows manually and by using our approach. Major result shows that our workflow integration and validation approach can support the work of the project manager in monitoring and comparing the designed workflow model and the running processes in heterogeneous engineering environments.

The remainder of this paper is structured as follows. Section 2 presents related work on heterogeneous engineering environments and workflow validation approaches. Section 3 identifies the research issues. Section 4 describes the solution approach. Section 5 presents a snapshot of a hydro power plant engineering project as a prototypic implementation for workflows integration and validation. Finally, section 6 concludes and identifies future work.

2 Related Work

This section summarizes background information on heterogeneous engineering environments and the basic concepts of the workflows validation approach.

2.1 Heterogeneous Engineering Environments

Research on heterogeneous engineering environments is quite new, while demands on using of software as a part of complex systems together with other engineering

[1] `http://www.processmining.org`

domains are inevitable [1]. There are some approaches for technical integration of component-based industrial automation systems. However only little work is available regarding the effective and efficient integration of engineering tools and systems along heterogeneous engineering processes.

Biffl et al. [1] propose an "Engineering Service Bus" to bridge technical and semantic gaps between engineering processes, models, and tools for quality and process improvements in software and systems engineering. The EngSB applies proven concepts of the "Enterprise Service Bus" from a business IT context to automation systems engineering. The major benefits of the EngSB (see the Open Source prototype implementation[2]) for heterogeneous engineering environments are as follows. (a) simple aggregation of components and services according to the project needs based on a common abstract infrastructure for communication between tools and systems, (b) improved coordination between tools that were not designed to cooperate by access to data and relevant changes in other tools, (c) legal recording and systematic closing of open loops in engineering team processes. However, further configuration and administration like validation and verification of process models need to be investigated, since the EngSB is only a middleware layer concept and does not provide additional advanced applications out of the box.

2.2 Workflow Validation

Some works on workflow validation, for example are done by Cook and Wolf [4]. They suggest for a process validation method based on techniques for detecting and characterizing between a formal model of a process and the actual execution of the process. These techniques are neutral with respect to the correctness of the model and the correctness of the execution. However, this work is more focused on single engineering systems. We need to expand this approach, so that it can handle different processes in heterogeneous engineering environments.

Sadiq *et al.* [7] propose a data flow and validation approach to address important issues in workflow modeling and specification. They identify and justify the importance of data modeling in overall workflows specification and verification. Their contribution includes and illustration and definition of several potential data flow problems that, if not detected prior to workflow deployment may prevent the process from context execution, may execute the process using inconsistent data or even lead to process suspension. However, the focus of their validation approach is more on the data validation, rather than on process validation of the workflow model. But the workflow also consists of processes that are required to be analyzed and validated.

3 Research Issues

Based on the requirement to manage heterogeneous workflows in distributed engineering environments, we identify the following two major research issues.

[2] `www.openengsb.org`

RI.1) Workflow integration approach. The integration of heterogeneous workflows is an important step to be able to manage heterogeneous workflows in the distributed engineering environments. Project managers need to have the big picture of the whole workflows under their supervision to be able to monitor and control the status of running projects of the systems.

RI.2) Workflow validation approach. The validation of workflow models with running engineering processes is a key issue to assess the quality of systems. Project managers want to be able to validate the integrated workflows with the engineering processes from different engineering fields, which are usually kept in event logs. The results of the validation could be a justification for project managers to improve the engineering processes of systems.

4 Solution Approach

This section presents: (a) the heterogeneous workflows in hydro power plant engineering as our use case for distributed engineering environments, (b) the framework for integrating and validating heterogeneous workflows from different engineering fields.

4.1 A Power Plant Workflow Management System

A power plant system, as a case of distributed engineering environments, consists of heterogeneous workflows that are controlled by different stakeholders. One type of such workflows could be a signal change management process, as described by Winkler et al. [11].

In power plant engineering systems integration, signals [9] are considered as common concepts in this domain to link information across different engineering fields. Signals include process interfaces (e.g., wiring and piping), electrical signals (e.g., voltage levels), and software I/O variables. We use signals as a vehicle to link domain-specific data between different engineering disciplines and introduce the application field “signal engineering”.

Important challenges that should be faced in managing signals change across disciplines are including (a) make signal handling consistent, (b) integrate signals from heterogeneous data models/tools, and (c) manage versions of signal changes across engineering disciplines [11].

For managing these signal changes, some workflows can be generated by different stakeholders according to their perspectives on how to accept/reject the signal change requests from other engineering fields.

4.2 Framework for Integrating and Validating Heterogeneous Workflows

To guide the process of integrating and validating heterogeneous workflows in the distributed engineering environments, we propose a framework illustrated in Figure 1. The framework shows two stakeholders and one project manager as users of this system. We can include more stakeholders to show that our approach can be applied for more stakeholders.

We divide the period of the system into two times, the design time and the runtime. In the design time, the stakeholders can create their own workflow models by using their favorite tools and notations, e.g., BPMN or Petri Net.

Those heterogeneous workflows are then integrated by using the EngSB approach [2]. The EngSB approach integrates different workflows by using the concept of the so-called Virtual Common Data Model (VCDM) that doesn't store data from each workflow physically, while it becomes a foundation for mapping proprietary tool-specific engineering knowledge and more generic domain-specific engineering knowledge to support transformation between related engineering tools [9].

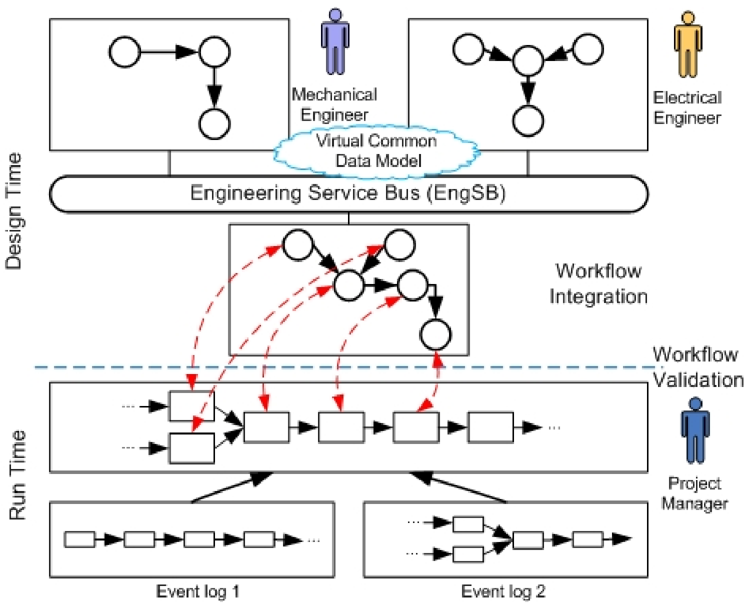


Fig. 1. Workflow Integration and Validation Framework

It is called "virtual" since actually there is no need to provide a separate repository to store the common data model. The management of the common data model with respect to different engineering fields is done via a specified mapping mechanism. The mechanism of the VCDM approach includes 5 steps: (a) Extraction or workflow data from each engineering field; (b) Storage of extracted workflow data into its own model; (c) Description of the knowledge for each engineering field's workflow; (d) Description of common domain knowledge; (e) Mapping of workflow knowledge to the common domain knowledge.

In the runtime, the operators control the running system and capture each event generated by each component of the system. The events later are stored and integrated in the event logs. The event logs are the foundation for workflow validation [8]. Project managers can use the information stored in the event log to validate the workflows that are already integrated in previous steps.

We use ProM, a process mining workbench for model discovery and conformance checking that supports workflow validation based on captured events [10]. Workflow validation is useful as justification for project managers in improving the engineering process quality.

5 Prototype Implementation

This section presents the scenario and the prototypic implementation of the change management workflow based on real world-data from a project at our industry partner, a hydro power plant system integrator.

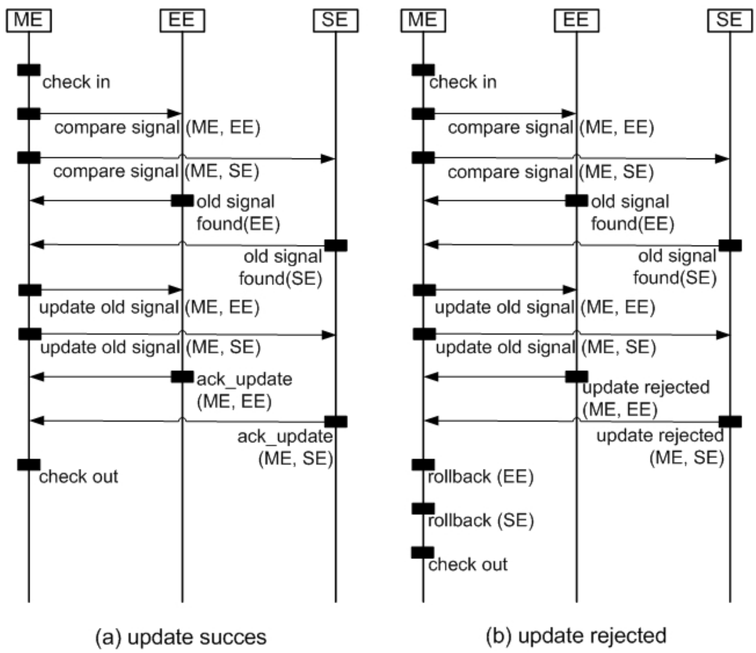


Fig. 2. Signal Change Management – Manual Scenario

5.1 Manual Signal Change Management

To show the improvement, that we have done at our industry partner in the hydro power plant systems integration domain, first we show the manual scenario of signal change management that is usually done between different engineering fields. In this scenario, there are three different engineering fields, namely mechanical engineering (ME), electrical engineering (EE), and software engineering (SE). The aim of this scenario is to have a consistent signal connection between different engineering fields. Usually, prior to the signal change, each engineering field, for example from the mechanical engineering, defines its connections to other engineering fields, for example to electrical engineering and software engineering. These connections are useful for cooperation between different engineering fields, for example some mechanical/physical switches are connected to electrical signals that can be controlled or monitored via software variables.

If one signal in the mechanical engineering is changed, other signals in other engineering fields also should be changed as well, to make the connection consistent. Figure 2 shows two scenarios on how to deal with and disseminate the signal change from one engineering field (ME) to other engineering fields (EE and SE), the first scenario (a) is when the signal updating success, and the second scenario (b) is when the signal updating fail to be spread to other engineering fields.

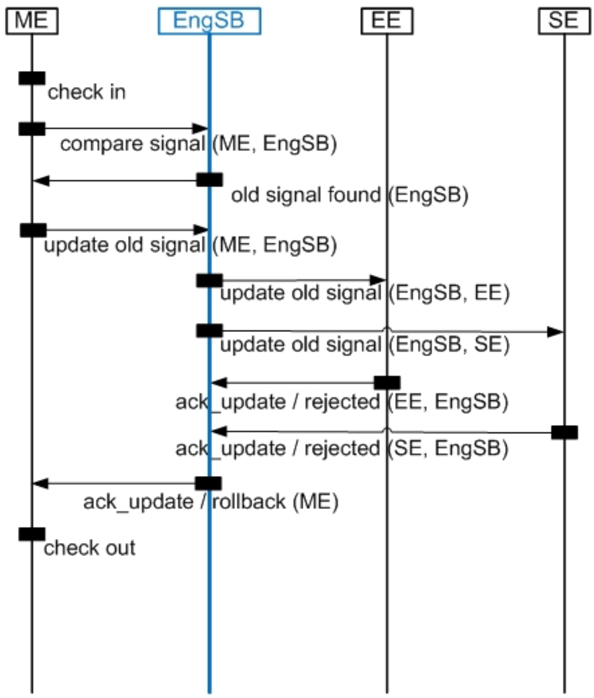


Fig. 3. Signal Change Management – Automated Scenarios using EngSB

First, the mechanical engineer (ME) checks in the signal changes in his work place. Then the ME will compare the signal to EE and SE. If related signals are found, the EE and SE will return the old related signals to the ME, such that the ME can update the relationships from old signal to EE and SE. If the update succeeds, the EE and SE will send acknowledgment to the ME. After completion of updates, the ME will check out the scenario. If the update failed, the EE and SE reject the update and this also will be sent to the ME. In this case, the update will be rolled-back from the EE and SE before check out.

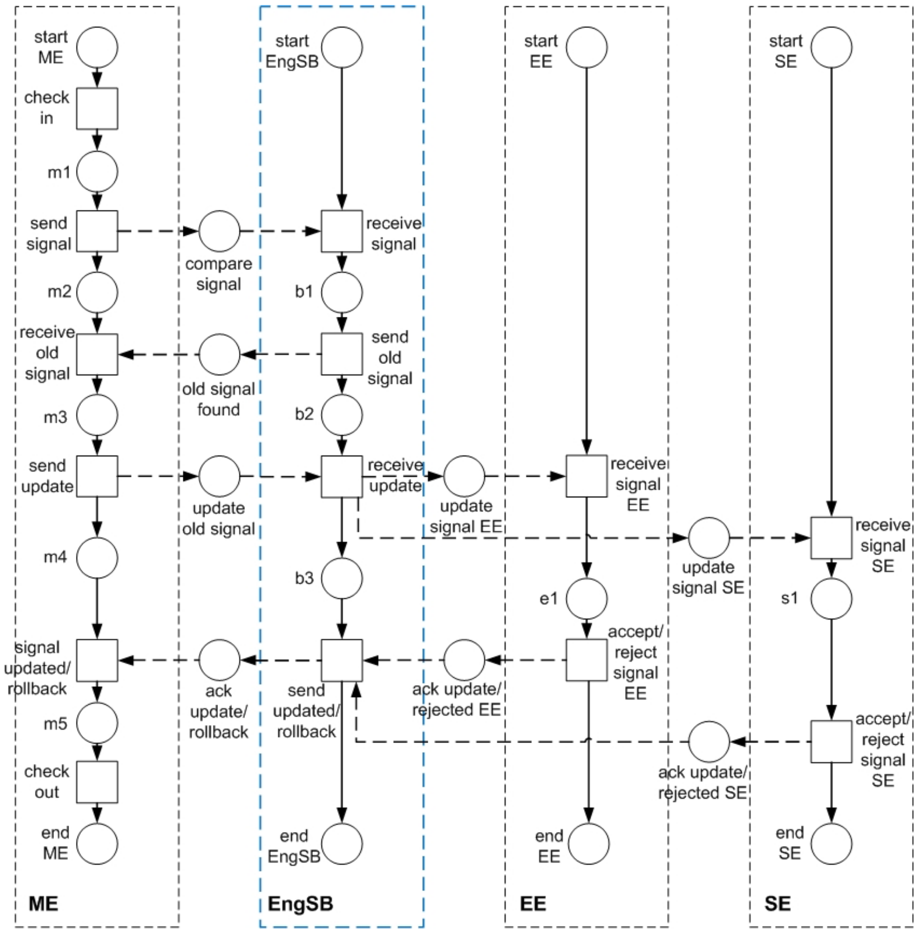


Fig. 4. Implementation of Workflows

5.2 Automated Signal Change Management

Automated signal change management, as illustrated in Figure 3, improves the situation described in the manual scenario, by adding the EngSB as a component to integrate and facilitate signal change management. In this case, the EngSB has a mechanism called VCDM to accommodate the signal matching from one engineering field to other engineering fields.

For example, the signal comparison is done only between the ME and EngSB, since all information from the EE and SE are already kept in the EngSB. So the ME only deals with the EngSB for signal comparison and gets feedback from the EngSB that the old signal has been found. The updating process is also simpler, since the ME only has to send the update request to the EngSB and gets a notification whether the update is succeed or failed, so the ME should rollback its signal changing. The EngSB has the responsibility to disseminate the signal change to other engineering fields, and sends a notification whether the update is succeed or failed.

5.3 Implementation of Workflows into Petri Net Diagram

In order to validate the heterogeneous workflows of signal change management in the power plants, we implement the signal change automated scenario using a Petri Net Diagram, as illustrated in Figure 4. The classical Petri Net is a directed bipartite graph with two node types called places and transitions. The nodes are connected via directed arcs. Connections between two nodes of the same type are not allowed. Places are represented by circles and transitions by rectangles [5].

The signal update is started by the ME. He checks in the signal update and sends a signal comparison request to the EngSB. The EngSB receives the signal comparison request, identifies the signal in its VCDM and sends back the old signal to the ME. The ME will send the update request to the EngSB that will be disseminated to the EE and SE. After getting replies from the EE and SE via the EngSB, the EE will send his acknowledgement that the signal has been updated or the signal update requests are rejected, in which case he should roll back the signal change in his engineering field.

The workflows implementation using Petri Net diagram is quite clear to show the scenario of signal change management from mechanical engineer to other engineering fields. This could help the project manager to learn the big picture of signal change management and the interaction between the engineering fields during the signal change processes.

6 Conclusion and Future Work

Integration and validation of heterogeneous workflows from different engineering fields are critical issues in distributed engineering environments, since individual disciplines apply different tools and workflow notations to represent their activities. This heterogeneity hinders efficient collaboration and interaction between various stakeholders, such as mechanical, electrical, and software engineers. In this paper, we have provided a framework to enable different workflows integration and validation. However, we still need more empirical studies to justify the benefit of this approach over other approaches.

Future work will include (a) improvement of process quality based on the workflow validation results, (b) application of other approaches to validate the workflows, (c) organizational analysis approach to check the source of change in the workflow.

Acknowledgments. This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria. We want to thank the domain experts at our industry partners for providing their expertise in discussing the processes described in this paper.

References

1. Biffl, S., Schatten, A.: A Platform for Service-Oriented Integration of Software Engineering Environments. In: Conference on New Trends in Software Methodologies, Tools and Techniques (SoMeT 2009), pp. 75–92. IOS Press (2009)
2. Biffl, S., Schatten, A., Zoitl, A.: Integration of heterogeneous engineering environments for the automation systems lifecycle. In: 7th IEEE International Conference on Industrial Informatics (INDIN 2009), pp. 576–581 (2009)
3. Chappell, D.: Enterprise Service Bus - Theory in Practice. O'Reilly Media (2004)
4. Cook, J.E., Wolf, A.L.: Software process validation: quantitatively measuring the correspondence of a process to a model. ACM Trans. Softw. Eng. Methodol. 8, 147–176 (1999)
5. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE 77, 541–580 (1989)
6. Raedts, I., Petkovic, M., Usenko, Y.S., van der Werf, J.M.E.M., Groote, J.F., Somers, L.J.: Transformation of BPMN Models for Behaviour Analysis. In: 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS-2007), in Conjunction with ICEIS 2007, pp. 126–137. INSTICC Press, Funchal (2007)
7. Sadiq, S., Orlowska, M., Sadiq, W., Foulger, C.: Data flow and validation in workflow modelling. In: 15th Australasian Database Conference. Australian Computer Society, Inc., Dunedin (2004)
8. Sunindyo, W.D., Moser, T., Winkler, D., Biffl, S.: Foundations for Event-Based Process Analysis in Heterogeneous Software Engineering Environments. In: Euromicro Conference on Software Engineering and Advanced Applications, pp. 313–322. IEEE Computer Society, Lille (2010)
9. Sunindyo, W.D., Moser, T., Winkler, D., Biffl, S.: A Process Model Discovery Approach for Enabling Model Interoperability in Signal Engineering. In: 1st Workshop on Model Driven Interoperability (MDI), Oslo, Norway (2010)
10. van der Aalst, W.M.P.: Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing. RE Journal 10, 198–211 (2005)
11. Winkler, D., Moser, T., Mordinyi, R., Sunindyo, W.D., Biffl, S.: Engineering Object Change Management Process Observation in Distributed Automation Systems Projects. In: 18th European System and Software Process Improvement and Innovation (EuroSPI 2011), p. 12. Roskilde University, Denmark (2011)

Use Cases and Object Modelling Using ArgoUML

Wojciech Complak[1], Adam Wojciechowski[1], Alok Mishra[2], and Deepti Mishra[2]

[1] Institute of Computing Science, Poznań University of Technology, ul. Piotrowo 2, 60-965 Poznań, Poland
[2] Department of Computer Engineering, Atilim University, Incek, 06836 Ankara, Turkey

Abstract. The key factor in the process of good quality software development is using proper techniques for requirements elicitation, representation and modelling providing foundations to build a logical model of the projected system.

One of the most popular functional requirements description methods is use cases and scenarios approach introduced by Jacobson in 1993. Use case diagrams and analysis of functionality necessary for future users of the system allows precise definition of object model of target software system. However there are few fully functional free CASE tools supporting system designer at this software development stage.

The article describes process of creating guidelines and formal requirements for design of a software system supporting storage and analysis of data concerning regulatory duties of Polish Telecom (Polish name: Telekomunikacja Polska) in the range of RIO framework agreement regarding collocation with the free UML modelling ArgoUML tool.

Keywords: UML, Software Engineering, use cases, open source software.

1 Introduction

In Poland the construction of telecommunication infrastructure was started between World Wars by a national company Polska Poczta, Telegraf i Telefon (PPTiT: Polish Post, Telegraph and Telephone) established in 1928 [1]. The ongoing efforts to modernise and develop the existing infrastructure and to reorganise services towards market and commercial needs were interrupted by the beginning of the Second World War in 1939.

After World War II PPTiT was a de facto monopolist on the market. This dominance position began to change in 1987 when the Lower Chamber of Polish Parliament started fulfilment of the second stage of the economic reform. The primary target of the reform was basic reconstruction of the national economical system. Therefore PPTiT was transformed into a multi plant company based on market economical rules. Finally in 1991 Telekomunikacja Polska S. A. (Polish Telecommunication JSC, Polish Telecom in short) was span off as an independent company. Initially the company was a monopolist, nowadays it remains a

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 246–255, 2011.
© Springer-Verlag Berlin Heidelberg 2011

dominating operator on the market of telecommunication services in Poland. The task of supervising of elimination of national monopolies and preserving competition in the telecommunication services area in Poland was initially assigned to Urząd Regulacji Telekomunikacji (Telecommunication Normalization Office), later to Urząd Regulacji Telekomunikacji i Poczty (Telecommunication and Post Normalization Office) and since 2006 to Urząd Komunikacji Elektronicznej (UKE: Electronic Communication Office).

The paper presents experience and conclusions drawn while developing a software system project supposed to support control duties of UKE related to Polish Telecom in the range of collocation areas sharing. The original project was scheduled to be finished by the end of 2007 but it was slightly (January 2008) delayed and has been recently revisited to verify its correctness, to perform validation of assumptions and software choice.

Polish Telecom based on appropriate provisions of Polish law must provide among other telecommunication services the so called collocation service. Collocation service consists of sharing physical area and/or technical devices necessary to place and connect required equipment of an operator joining its network to other operator's network.

All document circulation related to collocation service has been performed in paper. In order to supervise fulfilling of the collocation duty by Polish Telecom it is necessary to build a software system providing means to gather and analyse appropriate data. To achieve this aim UKE shall order Polish Telecom to create a suitable computer system replacing current circulation of paper documents. The projected system is supposed to be able to exchange data with existing legacy inventory systems like SEZTEL or accounting and billing systems like SERAT/SERAT2.

The purpose of the commissioned by UKE task was to elaborate guidelines and formal requirements for design of a software system capable of storing and delivering information about collocation areas provided by Polish Telecom to other telecommunication operators. The projected system shall also have ability to automatically generate and deliver notices about violation of enforced by law time limits of performing statutory obligations by Polish Telecom.

The project comprised a range of introductory activities including analysis of existing regulations and applied practices and subsequently tidying up and formalising identified procedures.

The project covered the following detailed tasks:

– reviewing existing RIO framework agreement in order to tidy up laws and duties of telecommunication operators and Polish Telecom
– working out standards and rules to be applied by Polish Telecom while realizing collocation service including equipment of collocation areas:
  • elements of telecommunication devices (racks, stands, switches),
  • construction elements (for placing cable ducts)
  • auxiliary elements and systems (air conditioning, ventilation, heating, power, lights, grounding, alarm, fire control, visual supervision, access control)

- analysing applied imprecise informal and hardly documented procedures, defining and formalising them; procedures subject to analyse included:
  - concluding and cancelling Collocation Contracts
  - concluding and cancelling Detailed Collocation Contracts
  - accessing information about the number and location of collocation areas shared by Polish Telecom
  - access to information about collocation services rendered by Polish Telecom
  - reserving resources
  - releasing resources
- defining (as automated as possible) control and statistical functionality providing supervising office (UKE) information about:
  - current status of collocation areas shared by Telekomunikacja Polska and the information about possibility of co-usage of the infrastructure by another telecommunication operator
  - statuses of individual applications for collocations
  - violations of time limits related to procedures

All the others detailed tasks were related only to the software system and included:

- defining functional requirements for the future software system – the principal selected use cases method [2] for this task
- defining logical data model basing on the analysis of existing paper documents and preliminary approximation of data size to estimate type and cost of storage necessary to gather and archive data
- creating a software system object model with a commonly used software tool supporting data exchange in XMI (XML Metadata Interchange) format [3]
- elaborating software system interface allowing entire removal of paper document exchange between telecommunication operators and Telekomunikacja Polska
- defining requirements for Telekomunikacja Polska to assure software system safety, availability and maintenance
- proposing projected system architecture based entirely on free open source tools and public data exchange standards
- elaborating of data exchange interfaces between existing legacy systems (i.e. SEZTEL and MARTA) and projected software system
- constructing a procedure of selecting future implementer of the system

The commissioned task comprised using scenarios and use case for functional requirements analysis stage and object oriented analysis for modelling stage. As the comparison of popularity of different techniques ([4]) used in these two stages indicates that scenarios and use case are a quite popular method of requirements analysis (over 50% of popularity). Object oriented analysis which is supposed to be used along with use cases appears to be not as popular as one should expect it to be. Anyway the emphasis on object oriented technology in tools languages and books in the industry is very strong.

Unexpectedly the process of selecting proper tools for the project appeared to be not that straightforward. While the UML Unified Modelling Language [5] appeared to be the only reasonable solution (well defined standard, mature, concise, scalable formal language) the selection of tools turned to be more difficult.

The principal required well known, free open source tools supporting as many UML diagrams as possible and as flexible in supporting XMI format as possible. It is easy to notice (i. e. [6]) that basically there are numerous free UML modelling tools. However many of them are unfinished, unstable, orphaned or occasionally updated. Usually there is hardly any documentation available, and even if there is some it is at least outdated and incomplete. Besides it is difficult to precisely define what means "well known" since there are no public feasible statistics pertaining open source products (comparing for example to research presented in [4]).

Finally, after analysing capabilities of several tools and some discussion with the principal ArgoUML [7] version 0.24 dated 02-17-2007 has been selected (Fig. 1. contains a screenshot of the program with the outlook of the projected system). According to the creators of the program it supports UML 1.4 however we had some problems related to sequence diagrams.

Undoubtedly ArgoUML is a well-known application - from the 21 nominees, readers of the Software Development Magazine chose ArgoUML Software Design Productivity Award Winner (since 2006 Software Development Magazine merged Dr. Dobbs journal). The runners-up included well known commercial products for example Borland Together [8].

ArgoUML is written entirely in Java and it is open source BSD type licensed. As mentioned earlier ArgoUML does not support full UML standard (as far as types of diagrams are concerned) and it has limited editorial and visual capabilities.

After performing several data exchange tests between different programs we assessed that its support of XMI format is sufficient. Besides the principal suddenly valued its inherent portability – namely the ability of utilizing Java Web Start [9] technology allowing running ArgoUML on (virtually) any device supporting Java Runtime Environment (JRE) which is part of Java SE 6 platform [10].

Additionally it is possible to generate implementation in (among others) Java programming language directly from the model. During realization of the project we assumed – according to the developer statement ([7]), that in April 2007 there would be an SQL plugin available but apparently this subproject encountered problems because by the time of completing first stage of the project (March 2008) no such plugin or update of the schedule became available.

The most important advantage of using UML as a programming tool is the possibility of trespassing directly from the UML model to the application code thereby assuring that every aspect of previously verified and certified model will be implemented. This feature reduces implementation costs and increases its portability.

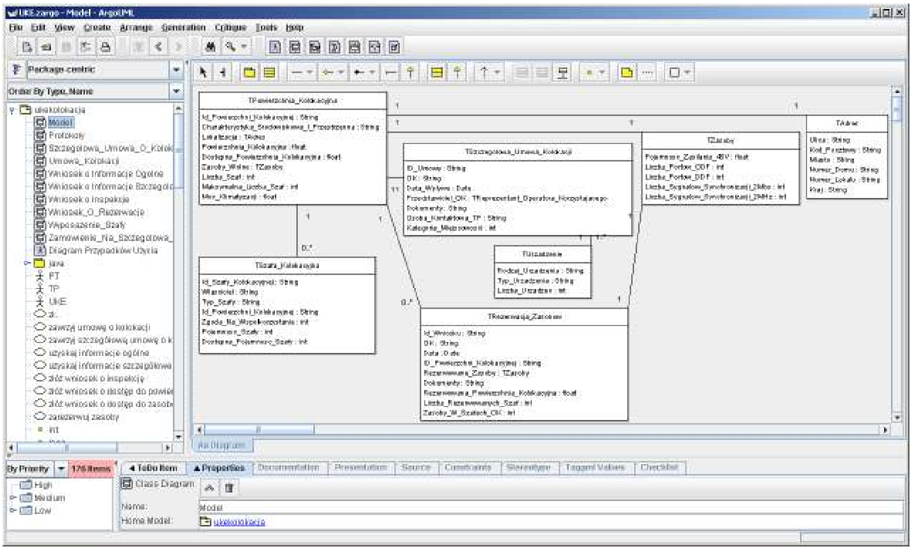


Fig. 1. The outlook of the projected system in ArgoUML version 0.24

Apparent disadvantages of Argo UML were – as mentioned earlier – incomplete implementation of UML standard and some functional inconveniences like for example lack of autorouting of diagrams, slowing in time, crashes causing loss of unsaved data.

Use case diagrams and use case scenarios creation was quite a different problem. While it is possible to create use case diagrams with ArgoUML (however without any significant benefit) it is not possible to create use case scenarios with it. Several tools like Use Case Editor [11] and Use Case Maker [12] were considered as an alternative however their application to this particular project yielded very limited profit.

2 Software Project Development

In the beginning appropriate use cases for formalised resource sharing procedures in the format proposed in [2] were developed. During this phase actors were identified and their roles in the project were defined. It appeared that the number of actors is relatively small and includes Polish Telecom (TP actor), Electronic Communication Office (UKE actor) and telecommunication operators (OK actor). The number of functions used by particular users was also relatively small. During the analysis process the following actors and their roles have been identified:

– UKE (*Electronic Communication Office*)
  Role: regulation, analysis and control of the telecommunication market
  Role towards the system: controlling proper sharing and managing of collocations
– TP (*Polish Telecom*)
  Role: telecommunication operator
  Role towards the system: construction, maintenance, sharing of collocation areas (final model by definition does NOT allow TP to be an OK)
– OK (*Operator Korzystający*)
  Role: entrepreneur or other subject legitimate to perform business activities according to separate regulations, performing business activity concerning supplying telecommunication network or telecommunication services
  Role towards the system: renting, using, sharing and terminating the collocation agreement

For all procedures identified and formalised during analysis phase there were adequate use case scenarios defined according to methodology described in [5]. An exemplary use case scenario is depicted in Fig. 2. All use case scenario were created using text processor with the aid of Microsoft Visio for drawing use case diagrams.

All uses case diagrams have been summarized in ArgoUML. It was easy to notice how much this tool is limited as far as effective and attractive presentation is concerned. It is only possible to export all diagrams to a bitmap format PNG. It is possible to export selected diagrams one–by–one to PNG, GIF, SVG, PS and EPS formats.

The number of defined use case was relatively small and comprised:

– submit an application for a collocation contract
– conclude a collocation contract
– conclude a detailed collocation contract
– acquire general information
– acquire detailed information
– submit an application for inspection
– submit an application for access to collocation area
– make resource reservation
– submit an application for resource access

In the next step an object model of the entire system (as a class diagram) was developed (an overview diagram depicting associations and attributes of classes is presented in Fig. 1).

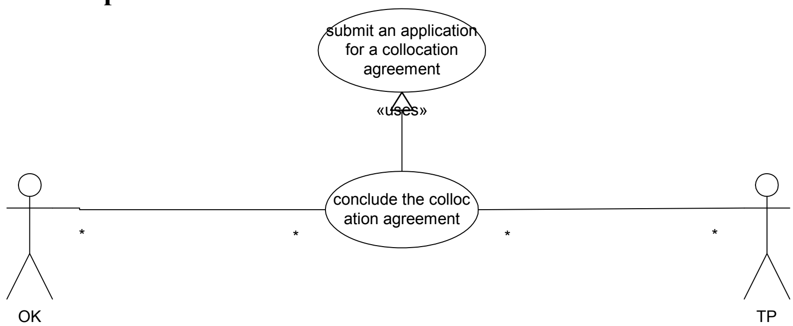
Basing on existing paper documents particular classes and their attributes were designed. Preliminary estimation of data size was performed according to implementation in Java generated by ArgoUML. An exemplary TUrzadzenie containing a device description class generated by ArgoUML follows:

Use Case: submit an application for a collocation contract

Brief Description

The use case initiates striving for a collocation contract by submitting an application for a collocation agreement

Relationships



Basic Flow

The use case starts when OK submits an application for a collocation agreement to TP.
The use case verifies formal correctness of an application for a collocation agreement, checking if all fields in the application have been properly filled in.
The use case recognizes application for a collocation agreement as formal error free.
The use case ends.

Alternative Flows

FORMAL ERRORS IN APPLICATION

The use case starts when OK submits an application for a collocation agreement to TP.
The use case verifies formal correctness of an application for a collocation agreement, checking if all fields in the application have been properly filled in.
The use case detects formal errors and summons OK to supplement the application/submit additional clarifications. The OK is presented a precise list of formal errors.
The use case supplements formal errors or submits additional clarifications from TP.
The use case recognizes application for a collocation agreement as formal error free (simplified verification procedure).
The use case ends.

Special Requirements

Unless TP verifies application for a collocation agreement within 3 work days the use case automatically recognizes application for a collocation agreement as formal error free.
Unless OK corrects all formal errors indicated in the call for supplementing application for a collocation agreement the use case resummons OK to supplement the application/submit additional clarifications.

Initial requirements

OK hasn't concluded a collocation agreement.

Fig. 2. An exemplary use case scenario

```
public class TUrzadzenie
{
   public String Rodzaj_Urzadzenia;    // the kind of the device
   public String Typ_Urzadzenia;       // the type of the device
   public int Liczba_Urzadzen;         // number of devices
   public TSzczegolowa_Umowa_Kolokacji
           myTSzczegolowa_Umowa_Kolokacji;
                             // binding to detailed collocation
                             // contract description class
   public TWyposazenie_Szafy myTWyposazenie_Szafy;
       // binding to description of rack equipment
}
```

Results were however very rough because it appeared that some of the documents contain scanned attachments (maps, authorizations, power of attorneys etc.) which size can significantly vary due to image quality and number of sheets of document.

As far as the definition of interfaces is concerned the project was totally unsuccessful. Polish Telecom refused to disclose any information about data structures or any interface to data stored in SEZTEL system. It was only possible do define general guidelines for implementer of the system. We also defined access rules for all types of subjects, backup plans, network attacks protections scheme and physical protection plans, safe communication protocols and schedule for regular system audits.

The proposed database system architecture is SOAP (a protocol for exchanging XML-based messages over computer networks) and free open source database tools like MySQL or Postgress. The user interface should be constructed with the aid of dynamic HTML pages. Some general remarks and hints useful during selection of future implementer of the system were also proposed.

3 Benefits of Using ArgoUML in a Distributed Team

Project managers responsible for choosing software tools to be used in the software process have to take into consideration several factors to achieve a compromise and optimal revenue [13]. An important issue is also introducing and involving customer representative into software team which has substantial positive influence on quality of communication and speed of software production [14]. Our experience of developing a software project in a distributed team showed that ArgoUML appeared a good alternative for other commercial tools. Freely available tool appeared to be known to all engineers and students who co-operated with us. Because many students used ArgoUML in their students projects they felt familiar with the tool and the overheads of introducing them into development environment was minimal. Our observation was that non-programmers, people with non-technical education were more likely to have some student experience with ArgoUML rather than e.g. IBM Rational Rose was considered as an alternative tool.

4 Conclusion

The paper presents experience and conclusions concerning realization of the contract commissioned by Urząd Komunikacji Elektronicznej in the range of elaborating guidelines and requirements for creation of a software system providing information about availability of collocation areas provided by Polish Telecom. The project has generally been successfully completed chiefly using free open source UML modelling tool ArgoUML in December 2007. The only detailed task which remained unfulfilled was defining data exchange interfaces between existing legacy systems and projected software system due to unwillingness of revealing relevant information by Polish Telecom.

The project was updated in summer 2011 to verify its validity and choice of tools. For many software projects it is vital to assure maintenance over several years. This problem was also considered during environment selection especially considering free software – one may expect that expensive commercial products sold by large companies will not perish easily. According to the same formal requirements it appeared that both Use Case Editor and Use Case Maker taken into consideration during initial selection were not suitable. Use Case Editor apparently is no longer developed – the last downloadable version is 4 years old. On the other hand Use Case Maker was significantly extended and could be useful if the project was to be started all over again.

As far as ArgoUML is concerned several improvements have been implemented over the years, eg.:

– new notation capabilities, previously only UML 1.4 and Java were available, now C++ support is added
– new reverse engineering features like importing from C++, Java from classes and Interface Definition Language IDL (previously only Java)
– rules library has been extended
– although interface has not changed significantly however it is more attractive (at first look one can notice colourful diagrams icons)
– ability to generate classes to source not only in Java (as before) but also C#, C++, PHP4/5 and SQL (the feature we were waiting for in 2007)
– the range of available diagrams has not changed significantly (still 7/14 of diagrams defined in UML), however there are more properties available (e. g. visibility)

The paper contains description of our experience gained during realization of the contract in the fields of methodology of analysing requirements and defining uses cases, conclusions concerning selecting a proper UML modelling tool and ArgoUML usage.

References

1. Poczta Polska, http://poczta-polska.pl/O_Firmie/?d=Historia&h=IIRzeczpospolita (in Polish)

2. Overgaard, G., Palmkvist, K.: Use Cases: Patterns and Blueprints. Addison-Wesley (2005)
3. The Object Management Group (OMG), http://www.omg.org/ (standard ISO/IEC 19503:2005 Information technology – XML Metadata Interchange)
4. Neill, C.J., Laplante, P.A.: Requirements Engineering: The State of the Practice. IEEE Software 20(6), 40–45 (2003)
5. Miles, R., Hamilton, K.: Learning UML 2.0. OâĂŹReilly (2006)
6. List of UML tools, Wikipedia (March 2008), http://en.wikipedia.org/wiki/List_of_UML_tools
7. ArgoUML, Tigris.org, Open Source Software Engineering Tools, http://argouml.tigris.org/
8. Borland Together, Visual Modeling for Software Architecture Design, http://www.borland.com/us/products/together
9. Sun, Desktop Java, Java Web Start Technology (2008), http://java.sun.com/products/javawebstart/
10. Sun, Desktop Java, Java Runtime Environment (2008), http://java.sun.com/j2se/desktopjava/jre/index.jsp
11. Use Case Editor UCEd (2007), http://www.site.uottawa.ca/~ssome/Use_Case_Editor_UCEd.html
12. Use Case Maker, http://use-case-maker.sourceforge.net/index.html
13. Mishra, D., Mishra, A.: A review of non-technical issues in global software development. International Journal of Computer Applications in Technology (IJCAT) 40(3) (2011)
14. Wojciechowski, A., Wesolowski, M., Complak, W.: Experimental Evaluation of 'On-Site Customer' XP Practice on Quality of Software and Team Effectiveness. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 269–278. Springer, Heidelberg (2010)

ORM 2011 PC Co-chairs' Message

Following successful workshops in Cyprus (2005), France (2006), Portugal (2007), Mexico (2008), Portugal (2009), and Greece (2010), this was the seventh fact-oriented modeling workshop run in conjunction with the OTM conferences. Fact-oriented modeling is a conceptual approach for modeling and querying the semantics of business domains in terms of the underlying facts of interest, where all facts and rules may be verbalized in language readily understandable by users in those domains.

Unlike entity-relationship (ER) modeling and UML class diagrams, fact-oriented modeling treats all facts as relationships (unary, binary, ternary etc.). How facts are grouped into structures (e.g., attribute-based entity types, classes, relation schemes, XML schemas) is considered a design level, implementation issue irrelevant to capturing the essential business semantics. Avoiding attributes in the base model enhances semantic stability, simplifies populatability, and facilitates natural verbalization, thus offering more productive communication with all stakeholders. For information modeling, fact-oriented graphical notations are typically far more expressive than other notations. Fact-oriented modeling includes procedures for mapping to attribute-based structures, so may also be used to front-end those other approaches.

Though less well known than ER and object-oriented approaches, fact-oriented modeling has been used successfully in industry for over 30 years, and is taught in universities around the world. The fact-oriented modeling approach comprises a family of closely related "dialects," including object-role modeling (ORM), cognition-enhanced natural language information analysis method (CogNIAM), and fully communication-oriented information modeling (FCO-IM). Though adopting a different graphical notation, the object-oriented systems model (OSM) is a close relative, with its attribute-free philosophy. The Semantics of Business Vocabulary and Business Rules (SBVR) proposal adopted by the Object Management Group in 2007 is a recent addition to the family of fact-oriented approaches.

Software tools supporting the fact-oriented approach include the ORM tools NORMA (Natural ORM Architect), ActiveFacts, InfoModeler and ORM-Lite, the CogNIAM tool Doctool, and the FCO-IM tool CaseTalk. The Collibra ontology tool suite and DogmaStudio are fact-based tools for specifying ontologies. Richmond is another ORM tool under development. General information about fact-orientation may be found at www.ORMFoundation.org.

This year we had 24 submissions for the workshop, contributed by authors from Algeria, Australia, Belgium, Canada, France, Luxembourg, Malaysia, The Netherlands, and the USA. After an extensive review process by a distinguished international Program Committee, with each paper receiving three or more reviews, we accepted the 12 papers that appear in these proceedings. Congratulations to the successful authors! We gratefully acknowledge the generous contribution of time and effort by the Program Committee, and the OTM Organizing

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 256–257, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Committee, especially Robert Meersman and Tharam Dillon (OTM General Co-chairs), and Pilar Herrero (OTM General Co-chair and OTM Workshops General Chair).

August 2011

Terry Halpin
Herman Balsters

ORM and Retirement of Outdated Business System

Chad Sevre, Necito dela Cruz, and Tor Westgard

Boston Scientific Corporation

Abstract. The use of Access 97 and Excel 2003 workbooks led to a high risk environment for the Global Supply functional pillar within Boston Scientific. Without making a change to the current system, Boston Scientific would not be able to implement a 3.6 million dollar savings per year cost reduction program in 2009. A retirement of complete collection of 55 Excel workbooks along with a 256 tiered Access 97 database system, incorporating dependent primary queries with many sub-queries, was required. To achieve the project goals, ORM was used to capture data requirements and all the related business rules. The ORM process showed the data gaps that were present in the current supply chain processes which allowed for a 20% mapping failure in the reporting. ORM also provided a more robust schema for product mapping in the reporting structure saving the company $300, 000.00 per year of the next five years.

1 Introduction

In an effort to implement an annual savings of 3.6 million dollar per year a corporate cost reduction program was introduced in 2008 for Boston Scientific Corporation (BSC). The corporate wide program was adopted due to a business change that affected how BSC would track inventory. Part of the business change involved implementing a new processing logic within SAP R/3 ("System Analysis and Program Development") for shipping freight to multiple distribution centers directly from the manufacturing plant that produced products being sold. Evaluations were done on down stream applications to understand the impact of implementing the corporate wide program. A major risk was identified for an existing Global Supply Chain application called "Report Launcher" which described its inability to provide reporting services to other functional pillars after a new IT development program called Replenish Direct was implemented. The Report Launcher application was comprised of Excel workbooks along with a multiple tiered Access 97 database system incorporating dependent primary queries with many sub-queries. There were 55 Excel workbooks and 256 Access databases. Without a major overhaul of the current database system, with its complex queries that transform the data, the reporting application could not be configured to deal with the modified data from SAP R/3. The project to replace the outdated supply chain system was called **D**ata **R**epository **Si**mplify **S**upply **C**hain **O**ptimization (DRSiSCO).

2 BSC Justification for DRSiSCO

The justification for DRSiSCO can be broken up into two items. The first was based on business need; be able to conform to the new changes being introduced into SAP

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 258–267, 2011.
© Springer-Verlag Berlin Heidelberg 2011

R/3. The second was a technical need; to have the business reporting system supported through the IS server infrastructure.

The current method for Global Supply Chain reporting was the use of discreet business managed Microsoft Access 97 databases integrated with the use of Microsoft Excel 2003 providing the reporting interface. The continuous use of these databases and workbooks over time eventually led to a system that:

1. is not scalable for increased data volumes
2. is not stable
3. does not have proper maintenance procedures
4. has complex reporting mechanism that is difficult to maintain

Of major concern was that the sheer volume of data to be passed from the modified SAP R/3 would exceed the functional limits of Access and Excel: 2GB limit for Access and the row limits for Excel. In addition to the multiplicity of dependent Access databases, the new logic would result in inefficient duplication of information storage and redundant data flows and interfaces for processing one report. Lastly, inventory data would be missed, because the SAP R/3 data would be passed through the reporting database and would not be able to account for multiple distribution centers' inventory of product being shipped directly from the manufacturing plant.

Figure 1 depicts the databases that comprise the existing reporting application, with the entire Parent databases in tier 1 linked to tables within the Child databases in tier 2

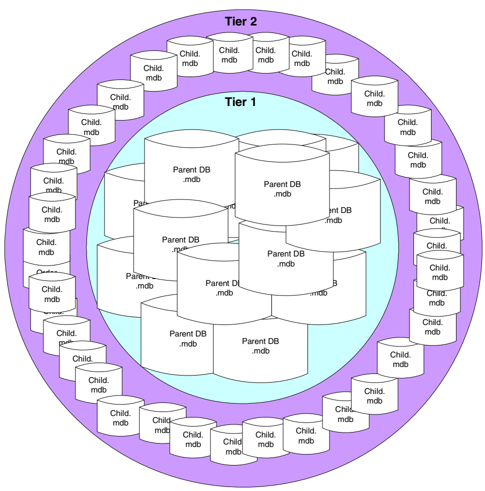


Fig. 1. Existing Global Supply Chain Reporting Application

With the risks identified, project DRSiSCO was initiated to investigate the impact of the SAP R/3, how the Global Supply Chain application might be modified, and/or how a replacement system could be built to support SAP R/3 changes.

3 Objectives for DRSiSCO

The DRSiSCO project first objective would lead to the retirement of the Global Supply Chain's current Access systems. This objective would provide the following results:

- Free up sever hard disk space for the business
- Reduce the amount of time to back-up the server each night
- Create efficient access to the same data
- Reduce the storage of redundant data

The second objective would lead to a centralized data repository that would provide the following results:

- Provide the ability for the Global Supply Chain group to restrict access to the data in a more controlled process
- Provide a stable platform for the amount of data that is being processed
- Provide the ability to scale the data to business needs
- Remove Access 97 query complexity thereby improving system performance
- Simplify design and implementation of future supply chain related projects

4 Data Quality and Good Models

Fundamental to achieving the objectives for DRSiSCO is to establish a "data quality strategy based on six factors – context, storage, data flow, workflow, stewardship, and continuous monitoring." [1] Moreover, because "there is a need to collect data from more than one original source, there are bound to be two specific issues that will potentially impact the effort: structural consistency, and semantic consistency." [2]

The use of canonical data models for both representation and for data sharing is the first step in enhancing the utility of the master repository. But these modeling approaches have to go beyond a simplistic relational map. Creative data modelers must understand the structures of their data sources as well as the semantics of the business terms reflected in each of the source models. The data model must make sense from a conceptual view as well as the logical and physical instantiations. Hierarchies and inheritance must be accurately reflected in the models along with the underlying chain of data element concept definition and element structure. And the models for representation must remain synchronized with the models for sharing and exchange. [3]

To find the most effective approach to achieving data quality project goals, ORM (Object Role Modeling) was leveraged to build a model of the repository that would reconcile structural inconsistencies as well as establish semantic consistencies, seek to rationalize the databases, simplify the data flows and interfaces, and establish a scalable, stable solution that would be subject to IS change control, documentation and maintenance procedures.

5 Timeline and Team Structure

Figure 2 summarizes the full timeline and key milestones of the project.

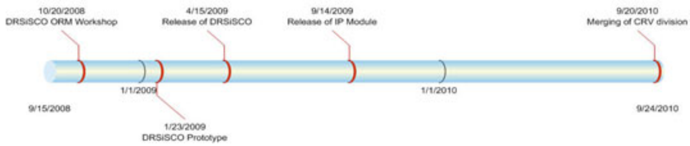


Fig. 2. Dr Sisco Project Timeline

The team was comprised of an IS Project Manager, IS BAs (Business Analyst), IS Data Architect, IS Programmers, Business Subject Matter Experts (SME) and a Technical Writer / Tester, as shown in the following diagram:

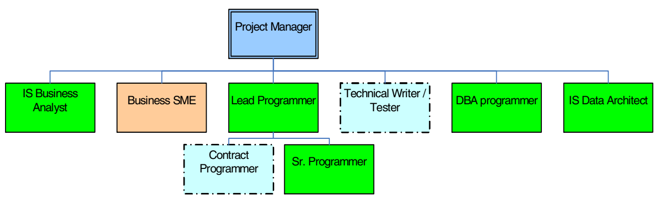


Fig. 3. DRSiSCO Team Structure

It was important to include all representative roles early on in the ORM process to ensure the overall database architecture would represent the business rules needed in capturing the data for the transformation logic. Many of the business rules were not documented by the business group because the old Access database system was built over many years. ORM was a new concept to the business functional pillar, Global Supply Chain, such that the SME's (Subject Matter Experts) were challenged to fully understand not only the complete flow of the data from SAP R/3 reports into the existing Access database system but also their meanings, relationships, and the business rules that were applied.

6 Architecture Analysis

To make sure the architecture was built for enterprise use, there was early engagement with the enterprise architecture (EA) group. They "provided enterprise architecture organization of logic for business process and IS infrastructure, reflecting the integration and standardization requirements of the firm's operating model". [4] An enterprise information architect from the EA Group assumed the key roles of Lead

Data Architect and Consultant in developing a data architectural framework. The as-is and to-be architecture states were defined to enable the team to create the logical design of the system.

7 As-Is and To-Be States

Figure 4, the as-is state, illustrates the different main core systems that supplied reporting data to the Global Supply Chain Group.

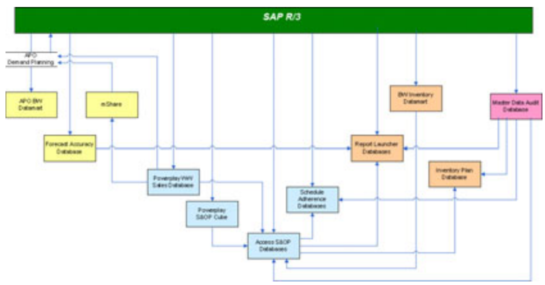


Fig. 4. As-Is-State of Supply Chain Data Flow

Figure 5 and Figure 6, the to-be state, illustrate the integrated data repository that supplies reporting data to the Global Supply Chain Group. From the simplified architecture diagram, the team recommended consolidating several databases into an integrated supply chain data repository. A subject area model representing major functional domains was developed that defined the boundaries of the ORM conceptual data modeling sessions as well as the timelines for roll out.

The Forecasting Reporting functional domain will be addressed as a future add-on to DRSiSCO.

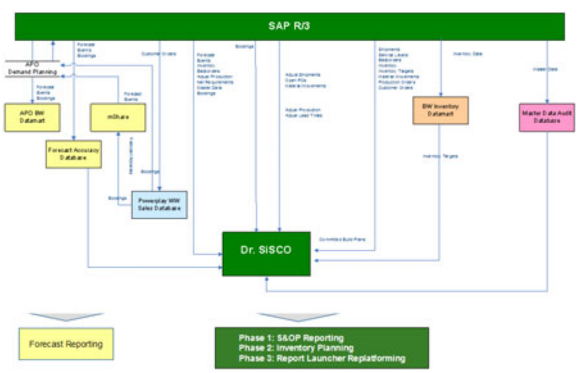


Fig. 5. To-be-State of DRSisCO

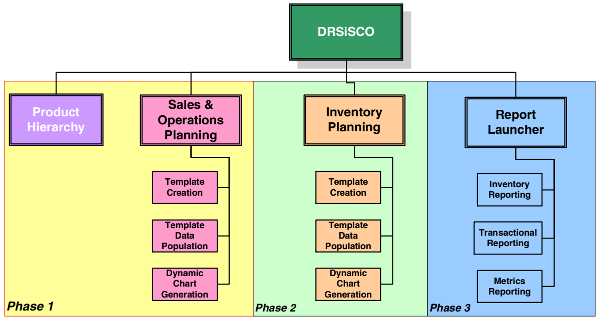


Fig. 6. Subject Area Model of DRSiSCO

The subject area model shows four different work streams: Product Hierarchy, Sales and Operations Planning, Inventory Planning and Report Launcher.

Product Hierarchy is used across all functional domains where planning, forecasting and reporting are performed at the different nodes of the corporate product hierarchy. This did not replace any databases but rather was a functional feature added to provide changes to the product hierarchy mastered in SAP for new products planned and forecasted to be manufactured and marketed. This was implemented in Phase 1.

Sales and Operation Planning or "S&OP" documents display the demand forecast, build plans, net requirements and inventory data in one place. This is where the company balances supply and demand. This replaced the following data sources: *Powerplay S&OP Cube*, *Access S&OP dat*abases, and *Schedule Adherence* databases. This was implemented in Phase 1.

Inventory Planning allows users to create planning documents with the goal of reducing inventory level across the company over time. It replaced the *Inventory Plan* databases. This was implemented in Phase 2.

Report Launcher is a separate reporting database that shares much of its data with DRSiSCO. It replaced the cumbersome reporting structure that existed in the old *Report Launcher* databases. This was partially implemented in Phase 3.

8 ORM Process

Prior to the project being officially approved, the IS Director for Manufacturing and Operations gave tacit permission to perform a series of conceptual data modeling sessions. The initial ORM session was held in Natick, Massachusetts, with the resident SME of Supply Chain Operations Planning (S&OP). A three day, 6 hour intensive ORM session was conducted and produced the base conceptual data models that were used to review with the St. Paul, Minnesota Supply Chain users.

When the project was officially approved, the Maple Grove, Minnesota DRSiSCO team was formed. The combined team started the process of ORM by having four hour sessions for two weeks with everyone on the team reviewing the ORM artifacts produced from the Natick sessions and identifying the business data feeds from each system. The data architect facilitated the review sessions and reconciled the differences between the Natick and Maple Grove versions.

With each feed, the team broke down the data into two workable sections. The first section was the review of the conceptual data model with the business users in Maple Grove. These reviews enabled the team to understand the relationships of the object types, their definitions, and the business rules that would be applied to the data for later use within the business reporting. The second section was for the project team, together with the business users, to define and understand the business rules applied when the data was extracted, transformed, and loaded into the physical database. The session information reviewed by the team discovered a data gap in the current process that allowed 20% mapping failure in the reporting process.

Figure 7 provides mapping examples that were developed during the ORM process.

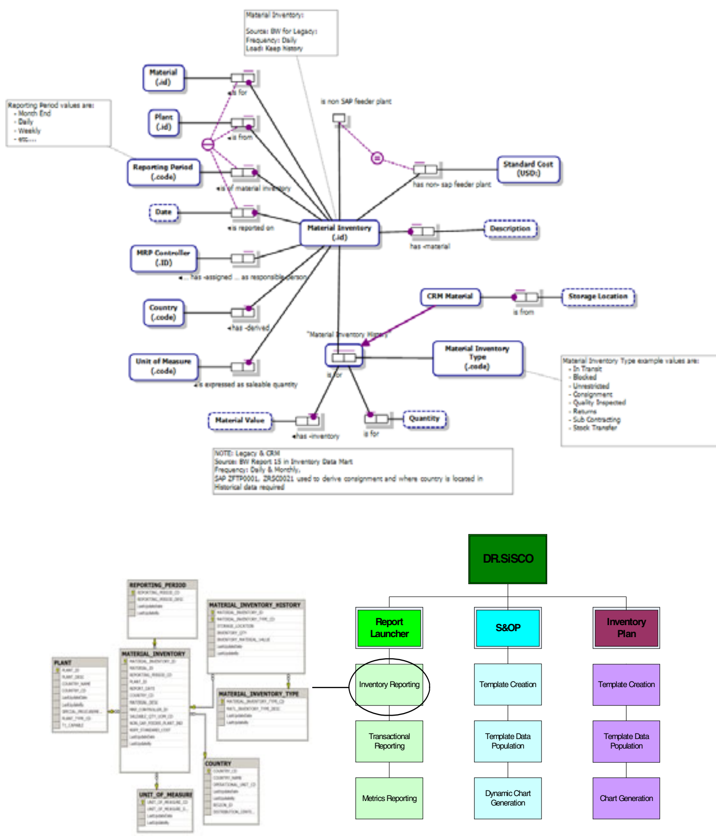


Fig. 7. DRSiSCO Inventory Reporting Schema Mapping Example

9 The ORM Result

DRSiSCO was the first application to incorporate ORM methodology within Legacy BSC. ORM methodology was absolutely critical in achieving the tight timelines. The first version was developed in a span of 9 months and implemented for BSC Legacy

Supply Chain business unit only. It subsequently went live prior to the Replenish Direct project in April 2009.

The base data model that was created, formed by 228 tables and approximately 1800 attributes, enabled for integration of another Supply Chain business unit – Cardio Rhythmic Vascular (CRV) in 2010. With the business rules already defined and applied to the data, this led to the ability to combine reporting of divisions dealing with inventory data for Boston Scientific. It was originally estimated that the integration of CRV with BSC Legacy Supply Chain would take months. But after performing ORM sessions to capture CRV specific Supply Chain reporting requirements, the estimate to merge the models was only one week. However, it was the conversion of CRV data into a common BSC Supply Chain process and reporting metrics that took time to clean, merge, and modify business processes. CRV data had to meet the quality standards based on the business rules, constraints and semantics defined through the ORM process.

The integrated data models provide highly stable and extendable schema for adding future functional domains that are planned to replace older applications within the supply chain system. A dramatic productivity and efficiency improvement have been realized in the length of time it takes for the S&OP administrator of the system to release approved sales and operations planning forecasts. The old system took the planning process a full 5 days of processing. DRSiSCO reduced work down to 1 hour of data processing time; this savings is broken down to an estimated $300,000 dollars in labor over 5 years, with a 3% return on investment for the business.

10 Conclusion: ORM Experiences

A post implementation 'lessons learned' meeting was conducted. Because ORM was new to the custom application group, a request was made to each team member to explain in their own words what they found to be beneficial of the ORM experience and their overall impression of the ORM work shops.

IT Business Analyst:

> "I find that the most powerful aspect of ORM modeling that I made use of as a BA is relating how data can or cannot be structured in a database to an inexperienced business user. Taking a simple case, when presented with the reality that a one-to-many relationship cannot function as a one-to-one relationship without issue, forces a business user to think differently about their data. ORM is the tool I use in these types of communications. This inevitably leads to a more robust data flow process and more reliable data for the end business user." - Tor Westgard

IT Manager:

> "ORM absolutely requires a strong partnership between the business SME and system/process SMEs. With multiple data objects and sources of data, both SMEs need to work together to understand the relationships and desired end state. The process of data modeling through ORM can also facilitate an understanding of complex data relationships and enable the SMEs to drive at

the end solution efficiently. I believe ORM modeling to be very applicable, in both reporting and in process solutions." - Noel Laguio

SME from the business pillar:

"My first impression of modeling was very confusing; it was difficult to understand & took me many months to try to come close to understanding. I don't think, based on my knowledge of process that I would ever attempt to do an exercise like this on my own. The end result is useful. " - Tami Eggert

Lead Programmer:

"The use of ORM in the modeling of data for the DRSiSCO proved to be a very valuable endeavor. The ORM process provided an excellent structure for interviewing the business users and documenting the results of the conversations into a workable model. The ORM process also provided a natural format for discovering the relationships and structure of the data elements required by the system. In comparison to other data modeling techniques I have used in the past I found that the ORM methodology worked much more effectively when dealing with a large and complex data model. The benefits of ORM were also very evident during the development phase of the project. The consistency of the data structures and relationships resulted in simpler database code and fewer defects during the coding process. In addition, the ORM process resulted in a set of documentation that was extremely valuable in training new resources joining the project. The process was successful in both reducing development costs and increasing software quality" - Jeff Sorenson

DBA Programmer:

"ORM is a disciplined data modeling methodology supported with a tool that can be used to decrease the development time and reduce the changes needed when developing a relational database for an application. It allows users to discuss and see the database objects in terms that similar to what they use every day. They can see the relationships between data structures, capture notes, and identify issues in a graphical format. The relationships that are displayed can be spoken back to the user to see if they capture what the user has described. Once everyone has agreed that the model is complete, it can be used to create the initial database for the application. As development of the application proceeds, there will be changes made to the database. These changes can be easily updated in the ORM model. After the application is complete, the ORM model can be used by new users and developers to help them understand the application. An ORM model is useful for the life of an application" - Steve Anderson

References

1. Business Objects, an SAP company, Data Quality Strategy: A Step-By-Step Approach. White Paper, p. 3 (2008)
2. Loshin, D.: Master Data Consolidation and Canonical Data Models. Knowledge Integrity, Inc., 2011 Sybase Article, v. 7.1, p. 1

3. Loshin, D.: Master Data Consolidation and Canonical Data Models. Knowledge Integrity, Inc., 2011 Sybase Article, v. 7.1, p. 2 – Good Models and Good Tools
4. Walker, D.M.: Resolving Data Warehouse Challenges Through Modeling. Sybase White Paper, pp. 1 Line 34 (March 2010)

A Metamodel for Orchestrating Service Requirements in an SQL-Environment

Baba Piprani[1] and Suneil Dham[2]

[1] MetaGlobal Systems, Canada
[2] Office of the Superintendent of Financial Systems, Canada
babap@attglobal.net, SDHAM@osfi-bsif.gc.ca

Abstract. Orchestration and Choreography often form the fundamental backbone for delivery of services in a Service Oriented Architecture (SOA). In a service based application environment, services need to be able to manipulate persistent data---usually in a stable structured SQL-environment. These service requests to data access services should not be one-of or tailor-made. They need to be reusable. The realization of the dynamic service orchestration requirements often necessitates a corresponding requirement for dynamic on-demand assembly of SQL-modules consisting of re-usable stored procedure modules, User Defined Functions, and including dynamic SQL stored procedures. An ORM/NIAM metamodel is shown that can support the realization of a "provider" model that is able to dynamically assemble the SQL-implementation components to form an SQL-implementation orchestration model to satisfy the dynamic requirements of the service based orchestrations. The implementation of this model is parameter driven, thus allowing provider component-sets to replace individually tailored modules for SQL data manipulation.

Keywords: SOA, ORM, SQL, Orchestration, Choreography, metamodel.

1 Introduction

This paper addresses a scenario in application development that is based on a Service Oriented Architecture (SOA) paradigm. Since such a "standard" SOA environment does not yet exist, this paper also sets the stage in which the applicability of the involved interfaces, components are thus defined in the proposed architecture.

In a typical business application scenario where there are say 50 User Interface screens addressing say 6 -10 major business functional areas—with each screen requiring some form of data manipulation in an SQL-database, it is typical to hand-code such data manipulation code for each data attribute set, again typically being sourced from multiple tables. The larger the size of the database, the more repetitive code that has to be hand-coded. What is more frightening is that the number of modules needed are sometimes equal to the number of SQL-tables—since each table or view may require a different search query expression. Thus a 400 table database may typically result in say 300 or so modules. The situation is exacerbated in that

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 268–277, 2011.
© Springer-Verlag Berlin Heidelberg 2011

different screens may have dissimilar data manipulation request on the very same table, thus dramatically increasing the total number of hand-crafted modules required.

In brief, the issues facing application development today is the constant re-building of silo type individualized code modules that only serve a single purpose. Some may argue this is the most efficient and productive method, as in manual assembly in a mechanical production line—a la the pre-industrial revolution era. In contrast, today's technology advancements in the mechanical assembly arena have brought about mass scale robotics in the assembly line. A parallel universe needs to be formed in the application development arena.

What this paper demonstrates is that it is possible to develop a parameter driven architecture containing a *single* metadata driven component set that is able to dynamically "stitch" together such required sub-components for the needed data manipulation requests—thus eliminating the need to hand-craft and develop individual data manipulation embedded modules that satisfy a given User Interface requirement. In other words, it is possible to create an "assembly line" to address data manipulation requests to be assembled for a "service".

The paper's focus is essentially on an architectural component that accesses a service to satisfy the data requirements of some pre-defined user interface component(s). The architectural and other requirements or the precise nature of the user interface component is not within the scope of this paper. Instead, the application and data access layers interface via a pre-established message format containing the semantics and nature of the user requirement as being generated from the user interface that is passed to the data layer. The return message from the data layer to the application layer is also similarly handled via a pre-established message format and content in response to the received request from the application layer.

2 Foundational Principles

Considering a service as a collection of capabilities bound together through a functional context, and overseen by a contract depicting its purpose and capabilities, the following foundational principles were taken into account in the design of the architecture shown in this paper. In brief: Standardized service contracts, service loose coupling, service abstraction, service reusability, service autonomy, service statelessness, service discoverability, service composability, and service oriented interoperability[1].

This paper essentially references the experience gained on a live project (RASP) at the Office of the Superintendent of Financial Systems (OSFI), Canada. The paper first defines the framework within which the canonical components are positioned in a collaboration diagram within the application and data layers that enable access and manipulation of structured data in an SQL-environment. The portrayal of this framework is done in two parts.

The first framework shows a collaboration diagram [2] portraying the simplistic approach that uses a data access component tailor made to address the data access and manipulation requirements for that particular service request being received from the application layer. An example of the code component module for SQL data-base access would be a pre-compiled stored procedure. Here the orchestration is hard-wired into the code module, albeit with the ability for realizing alternate paths and routings based on result parameters in a series of pre-conceived navigation steps.

The second framework portrays a scenario in a revised collaboration diagram that includes a master data hub, where a metamodel drives the dynamic assemblage of the orchestration requirements based on a service request requirement. Here a realization of this metamodel can be implemented from the contained semantics of the nature of the data access or manipulation sub-component. The metamodel semantics would also be able to drive the orchestration requirements in order to assemble an implementable component for data access and manipulation. The second framework approach would dramatically reduce the coding requirements for such data access and manipulation requirements as compared to the traditional tailor made one-module-for-each access and manipulation requirement.

2 The Simplistic Canonical Collaboration Model

Figure 1 depicts a simplistic version of a canonical collaboration model of four canonical collaboration views to represent the standard operations in a typical application: Create, Read, Update and Search, collectively referred to as CRUSH. For those more familiar with CRUD, it should be noted that given the Government of Canada policy on retention of data for seven years, there is no Delete operation, only an Update that sets the visibility of an item to logically deleted or archived. In the rare case when a true delete is an absolute necessity, e.g. erroneous data that negatively

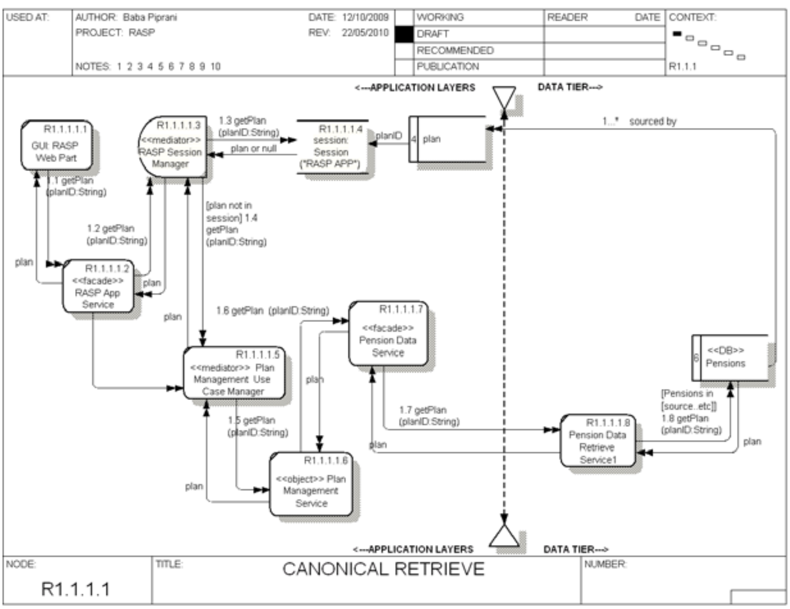


Fig. 1. Simplistic Collaboration View for Canonical Retrieve

impacts on the solution, this will be addressed by a SQL-database administrator or alternate user interface design components falling into routine exception handling, and is out of scope for this discussion.

The primary purpose of collaborations is to explain how a system of communicating entities collectively accomplishes a specific task or set of tasks without necessarily having to incorporate detail that is irrelevant to the explanation. It is particularly useful as a means for capturing standard design patterns.

As a result of the nth level decomposition of the Business Process Model, resulting in a leaf level elementary process [3], standard design patterns can be developed that could be slotted into CRUSH. A canonical collaboration view for each of these patterns is shown below. These canonical views can be used to develop suitable re-usable component services to satisfy the interface and data requirements as shown.

2.1 Collaboration View Canonical Retrieve

The RASP application canonical retrieve operation contains a number of components and objects across the n-tiered architecture of the application, as shown in Fig. 1. This is a notional diagram and is not to be interpreted as an exact prescription for the components and classes that is to be used in an implementation. Depending on the relative complexity, number of classes, interfaces and methods, it may make sense to split or merge certain notionally defined components. For example, the <<mediator>> subject area User Case Manager and <<object>> subject area Service components could be merged if not warranted as distinct components. The components and objects in Fig. 1 are defined in Table 1.

2.2 Collaboration View Canonical Retrieve and Canonical Search

With a request to retrieve pension plan information for a specific pension plan that is initiated at 1.1 getPlan from the GUI:RASP Web Part component, the request proceeds to the RASP Session Manager, Plan Management Use Case Manager to determine the Plan Management Service and, using Pension Data Service in the Application Layers to result in a call for data services via 1.7 getPlan to Pension Data Retrieve Service1. This service is essentially a main component tailored to address a specific data access request for retrieving a pre-defined set of Plan information as per the requirements of the GUI: RASP Web Part and as communicated by 1.7 getPlan. This main data service component, (say a User Defined Function or a Stored Procedure UDF 'getPlan(planID)', contains the necessary (and generally pre-compiled) data service sub-component(s) for example, UDF1 for retrieval of specific Plan information from the PENSIONS database from table PENSION PLAN, UDF2 for retrieval of specific plan information from table REGISTERED PLAN, UDF3 for retrieval of specific plan information from table PLAN HISTORY, UDF4 for retrieval of information from table PROVINCIALLY REGISTERED PLAN ...and so on, to satisfy all the necessary screen elements for the use case via data access requests being returned via 1.8 getPlan.

The data return values being returned in response to the 1.7 getPlan request by the main component UDF or stored procedure could result in a single data value from a scalar function or a table data type from a table-valued function. So, in response to the request 1.7, the Pension Data Retrieve Service1 then returns either a denormalized row with embedded arrays or, XML data. These are implementation decisions to be determined in detailed design phases, as with the use of precompiled UDFs or stored procedures, or dynamic SQL stored procedures for the main component or sub-components. The implementation decisions apply to the multiple scenarios depicted here.

The canonical search is similar to the Canonical Retrieve in terms of service requirements and operations except it uses a partial string or pattern match on a selected set of Plan attributes to be used in determining the set of PlanID(s) and reverts to a regular Canonical Retrieve.

2.3 Collaboration View Canonical Create and Canonical Update

With a request to INSERT pension plan information for a specific plan initiated at 1.1 createPlan from the GUI:RASP Web Part component, the request proceeds to the RASP Session Manager, Plan Management Use Case Manager to determine the Plan Management Service and, using Pension Data Service in the Application Layers to result in a call for data services via 1.7 createPlan to Pension Data Create Service1. The Pension Data Create Service1 as the main data service component (say Stored Procedure USP 'createPlan(planID)') contains a pre-determined insert plan consisting of the necessary data service sub-components for example, USP10 for INSERT of specific Plan information to the PENSIONS database into table PENSION PLAN, USP20 for INSERT of specific plan information into table REGISTERED PLAN, USP30 for INSERT of specific plan information into table PLAN HISTORY, USP40 for INSERT of plan information into table PROVINCIALLY REGISTERED PLAN ...and so on, to satisfy all the necessary screen elements to be INSERTed for the use case. Note that the insert plan will need to necessarily follow database structure rules in ordering of INSERTs, based on referential integrity and other integrity constraints as defined in the SQL Schema.

Exception conditions handling will need to be addressed through the use of a Business Rules Service component, which would essentially trap and identifiy the appropriate integrity violations, if any were encountered during the create operation.

The incoming data values into the USP Stored procedure in 1.7, could pass either a denormalized row with embedded arrays or, XML data.

3 Master Data Hub Based Canonical Collaboration Model

The second framework portrays a scenario in a revised collaboration diagram that includes a master data hub, where a metamodel drives the dynamic assemblage of the orchestration requirements based on a service request requirement.

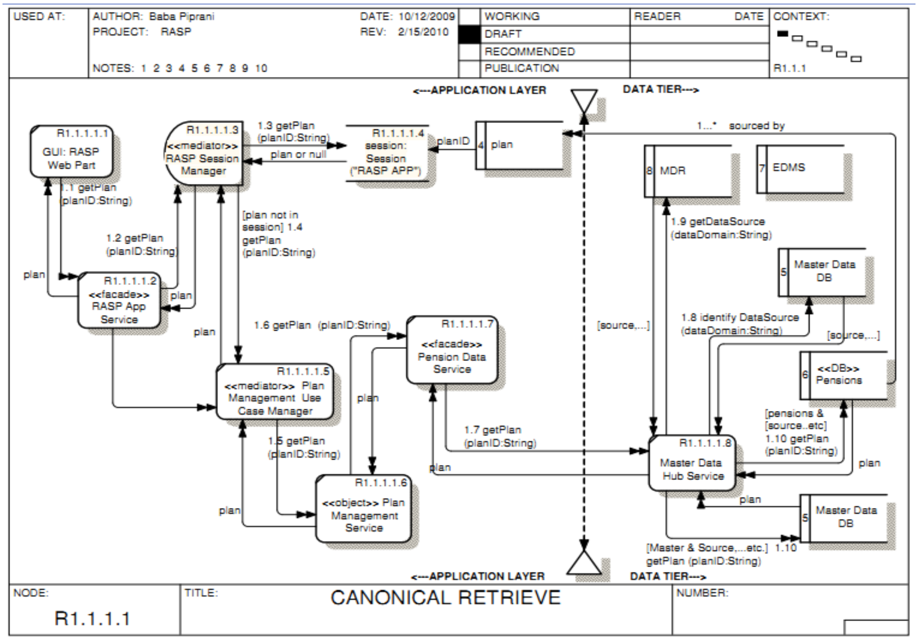


Fig. 2. Master Data Hub based Collaboration View for Canonical Retrieve

3.1 Master Data Hub Based Collaboration View Canonical Retrieve

The components and objects in Figure 2 are defined in Table 1.

Table 1. Component definitions for Master Data Hub based Collaboration View for Canonical Retrieve

Component	Description
GUI: RASP – Web Part	RASP application Graphical User Interface (GUI) component, a web framework component (e.g. an ASP.NET or other), likely a Web Part to permit seamless integration with a web collaboration and publishing platform (e.g.Microsoft SharePoint)
RASP App Service	Façade-pattern class that exposes consolidated component operations required by the RASP application graphical user interface
RASP Session Manager	Mediator-pattern class that manages a session cache of business data entity objects
Session	Cache of business data entity objects
Plan Management Use Case Manager	Mediator-pattern class that manages the collaboration of classes and objects that the given Use Case which is eventually decomposed into an elementary process as referenced above
Plan Management Service	Component of classes that fulfill the Use Case for Manage Plan Information. Depending on complexity of component model it may be appropriate to merge the use case manager and service components and classes

Table 1. (*Continued*)

Pensions Data Service	An item of value to an organization such as: printer, workstation, laptop, cell phone, smart phone, server, data center, USB drive, token or access card, software and physical media, strategic, policy and planning documents, personal information, financial information, asset inventories, security and risk assessment reports, information on IT network, architecture, IP addresses, system logs, password and account information
Master Data Hub Service	Enterprise Master Data Registry and Service that provides source, mappings and services for data manipulation including access control
Master Data Database	Enterprise-wide shared data database
Metadata Repository (MDR)	Master Data Registry for data attributes, mappings and services
Pensions Database	Private Pension Plans data database
STRATEGIC PARTNER	An external organization that can provide early warning, assistance, and advanced intelligence, or that the organization needs to communicate with, related to a security event such as: local emergency response teams (e.g. Fire Dept., Police…), local incident management teams (e.g. provincial emergency management…), local crisis management teams (e.g. Red Cross…), government (municipal, provincial, federal), service providers (e.g. security solution providers, financial/payroll, telecommunication…), forensic investigators, telecommunication, special protection groups (e.g. RCMP…), media (public communication)

3.2 Master Data Hub Based Collaboration View Canonical Retrieve and Search

With a request to retrieve pension plan information for a specific plan initated at 1.1 getPlan from the GUI:RASP Web Part component, the request proceeds to the RASP Session Manager, Plan Management Use Case Manager to determine the Plan Management Service and, using Pension Data Service in the Application Layers to result in a call for data services via 1.7 getPlan to Master Data Hub Service. The Master Data Hub Service consults the MDR with the planID to determine the method signature and attributes to identify the main data service component (say User Defined Function or Stored Procedure UDF 'getPlan(planID)') to be used, based on the mappings for methods against UDFs stored in the MDR. The data services component (say UDF getPlan (planID), based on a pre-determined access plan in MASTER DATA DB, then assembles the necessary data service sub-components for example, UDF1 for retrieval of specific Plan information from the PENSIONS database from table PENSION PLAN, UDF2 for retrieval of specific plan information from table REGISTERED PLAN, UDF3 for retrieval of specific plan information from table PLAN HISTORY, UDF4 for retrieval of information from table PROVINCIALLY REGISTERED PLAN ...and so on, to satisfy all the necessary screen elements for the use case.

The data return values from the UDF or stored procedure could be single data value from a scalar function or a table data type from a table-valued function. In resonse to the request 1.7, the Master Data Hub Service then returns either a denormalized row with embedded arrays or, a XML data.

3.3 Master Data Hub Based Collaboration View Canonical Create and Update

With a request to INSERT pension plan information for a specific plan initated at 1.1 createPlan from the GUI:RASP Web Part component, the request proceeds to the

RASP Session Manager, Plan Management Use Case Manager to determine the Plan Management Service and, using Pension Data Service in the Application Layers to result in a call for data services via 1.7 createPlan to Master Data Hub Service. The Master Data Hub Service consults the MDR with the planID to determine the method signature and attributes to identify the main data service component (say User Defined Function or Stored Procedure UDF 'createPlan(planID)') to be used, based on the mappings for methods against UDFs stored in the MDR. The data services component (say UDF createPlan (planID), based on a pre-determined insert plan in MASTER DATA DB, then assembles the necessary data service sub-components for example, UDF10 for INSERT of specific Plan information to the PENSIONS database into table PENSION PLAN, UDF20 for INSERT of specific plan information into table REGISTERED PLAN, UDF30 for INSERT of specific plan information into table PLAN HISTORY, UDF40 for INSERT of plan information into table PROVINCIALLY REGISTERED PLAN ...and so on, to satisfy all the necessary screen elements to be INSERTed for the use case. Note that the insert plan will need to necessarily follow database structure rules in ordering of INSERTs, based on referential integrity and other integrity constraints as defined in the SQL Schema.

Exception conditions handling will need to be addressed through the use of Business Rules Service component, which essentially traps and identifies the appropriate integrity violations, if any were encountered during the create operation.

The incoming data values into the UDFs or stored procedures could be single data value from a scalar function or a table data type from a table-valued function. The message to create in 1.7 to the Master Data Hub Service then would pass either a denormalized row with embedded arrays or, XML data.

The canonical update walkthrough is similar to the Canonical Create walkthrough in terms of service requirements and operations.

4 Metamodel for Orchestrating Service Requirements

The previous sections of this paper related to the establishment of a SOA application environment within which orchestration and choreography occur. So what do we mean by the terms orchestration and choreography in an SOA environment?

Much like the usage of "Services" and "Processes", there is a lot of confusion on the use of the terms Orchestration and Choreography in SOA. Simplistically, an analogy can be drawn to a conductor at an orchestra who directs the entire orchestra against a certain score as being Orchestration in SOA. Similarly, chamber music is an ensemble of soloists, essentially having one performer for each part, likened to Choreography in SOA.

Extending this analogy and to provide a consistent generic definition causes much confusion of terms, since there is much intermingling of the use of terms and semantics for a process vs. service. For example, a statement like "Orchestration refers to an executable business process that may interact with both internal and external Web services" [4][5] eludes that orchestration only applies to a business process that is executable. The Object Management Group (OMG) refers to orchestration as "the modeling of directed, internal business processes"[6].

Another way to look at it is that orchestration is described to be typically involving a given single swim lane as a controlled work flow, as compared to choreography that addresses collaboration across multiple swim lanes.

But there is widespread mixed usage of the terms and semantics for Services and Processes. Piprani, et al. [7] provide a clear separation of a Business Process, which then splits into an automatable process using the aid of computers and then decomposed into an atomic process, which, as an elementary unit of work is executable by a Service (again using computers).

In [7] it was also shown that there is a Service hierarchy and grouping. providing an overlay of positioning of Services and Processes. That paper also recognized that while Business Process Modelling is essentially a top-down process, the Services Oriented Architecture is a bottom-up process consisting of an assemblage of constituent atomic processes and/or services.

Fig. 3 depicts the ORM / NIAM orchestration metamodel for SOA. The entity PROVIDER is used to denote a SERVICE or an ATOMIC PROCESS. A SERVICE may be a COMPOSITE SERVICE or a LEAF SERVICE. A SERVICE has a hierarchy in that a LEAF SERVICE or a COMPOSITE may have a service that is a super. An ATOMIC PROCESS may call another ATOMIC PROCESS. SERVICE ORCHESTRATION occurs with an orchestrating service providing one or more PROVIDER services or processes in an orch step. This specific SERVICE ORCHESTRATION step may have an ORCHESTRATE CONDITION. An ORCHESTRATE CONDITION has a PRECONDITION and/or POSTCONDITION with a given EXPRESSION, OPERATOR (for meeting a result..e.g. Boolean value, equality) and related RESULT. The PROVIDER may have one or more parameters as in PROVIDER PARAMETER, denoting PARAM INOUT TYPE in a PARAM SEQ NBR. The PROVIDER PARAMETER must have a PARAM NAME and DATATYPE.

Thanks are due to Daryl Graf, John Calvert, Luke Jones, Omer Madi and Ryda Jubran for their lively discussions and input during the formulation of these models.

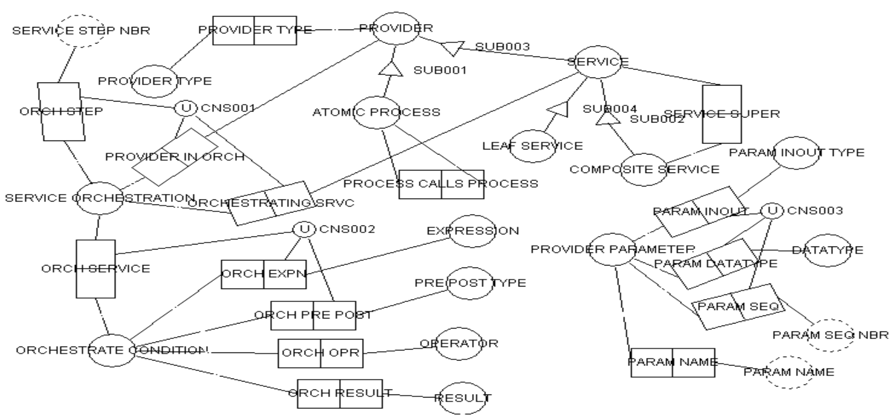


Fig. 3. ORM Metamodel for Orchestration in SOA

5 Conclusion

The ORM metamodel for Orchestration in SOA essentially enables automation of the Data Tier, and provides a dynamic assembly mechanism that can be implemented in an SQL-environment providing reusability and dramatic reductions in development.

References

1. Erl, T.: SOA:Principles of Service Design. Prentice-Hall Service-Oriented Computing Series, http://www.SOAbooks.com
2. International Standard ISO 19501 – Unified Modelling Language Specification, International Standards Organization, Geneva
3. Piprani, B., Borg, M., Chabot, J., Chartrand, É.: An Adaptable ORM Metamodel to Support Traceability of Business Requirements across System Development Life Cycle Phases. In: Chung, S., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 728–737. Springer, Heidelberg (2008)
4. Laube, A., Winkler, P.: Generation of Choreography Skeletons from Web Service Definitions, http://www.posecco.eu/fileadmin/POSECCO/user_upload/publications/WSCDL_final.pdf
5. Pelz, C.: Web Service Orchestration and Choreography, http://wpage.unina.it/rcanonic/didattica/at/documenti/wsOrchestration.pdf
6. SOA Orchestration and Choreography, http://searchsoa.techtarget.com/answer/SOA-orchestration-and-choreography
7. Piprani, B., Wang, C., He, K.: A Metamodel for Enabling a Service Oriented Architecture. In: Chung, S., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 668–677. Springer, Heidelberg (2008)

Enterprise Data Architecture: Development Scenarios Using ORM

Necito dela Cruz, Mary Tobin, Gabe Schenz, and Dick Barden

Boston Scientific Corporation

Abstract. This paper describes the enterprise wide use of ORM to drive individual project development through requirements, design, construction, and implementation process using SDLC practices within Boston Scientific. Components used to construct, track, administer, and manage data models using nORMa for conceptual data models and interfaces to an industry standard tool for the logical and physical data models are laid out in one of the three following scenarios: 1) New customized application development, 2) OTSS configuration and customization (Off the Shelf System), and 3) Reverse Engineering Existing Applications. This paper illustrates how the second scenario "OTSS configuration and customization" leverages Object Role Modeling to provide the data architect with a fact based conceptual model that is integrated with a business glossary. The ultimate goal is to provide a common understanding of the business at a conceptual level and at the same time provide a comprehensive and deep 'where used' capability that supports a variety of common functions for Data Architects at the Enterprise level and Data Architects at the Project level.

1 Introduction

This paper culminates a four year process of evolving an Object Role Modeling framework into an integrated architectural practice for Boston Scientific Corporation. During the past two years of 2009-2011, a core group of enterprise architects implemented the four domains of enterprise architecture: Business, Information, Technology (includes Application and Infrastructure domains). Figure 1 below shows the synergy between the high level IT functions and enterprise architecture.

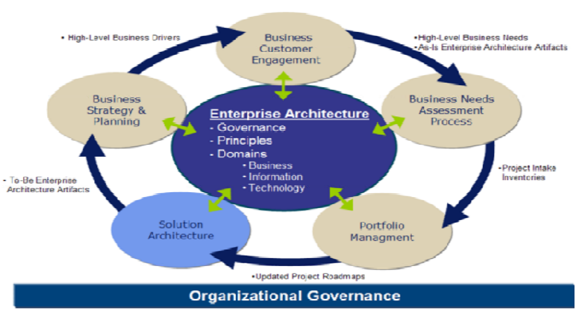


Fig. 1. BSC Enterprise Architecture Domains and Governance[1]

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 278–287, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Enterprise Architecture adds value by simplifying and adding uniformity to the environment in which collections of applications interoperate. Enterprise Architecture conveys its principles, concepts, designs and plans to others through artifacts, which in turn extends vision, increases speed and agility, refines insight, broadens reach, and dedication (i.e. improves operations) by developing and applying architectural practices as summarized in Table 1 below.

Table 1. Enterprise Architecture Value[2]

Enterprise Architecture Value	
Vision	Provides a mechanism to facilitate collaboration regardless of project, business unit, or location to deliver a solution with distinct value
Speed and Agility	Provides a proven technique for rapidly understanding available technology, integration options, data avail-ability/use, and supports a highly adaptable infrastructure
Insight	Ensures that the key decisions will be made at critical times during a project's lifecycle
Reach	Provides a means for communicating decisions across the enterprise; thus, facilitating solution reuse, extending project value and enterprise interoperability
Dedication	A common architecture will reduce project complexity and risk by establishing standards and principles

Specifically, the Information Architecture domain in BSC has sought to bring deeper semantic understanding to the collection of systems throughout Boston Scientific Corporation (BSC) by leveraging the features and capabilities of Object Role Modeling. Two previous papers delved into various aspects of this goal: 1) Object Role Modeling Enabled Metadata Repository (OTM 2007)[3]; and, 2) ORM and MDM/MMS Integration in an Enterprise Level Conceptual Data Model (OTM 2010).[4]

2 BSC System Development Scenarios

There are three major scenarios for system development and managing information models in BSC. They are: 1) New customized application (In-House) development; 2) OTSS (Off the Shelf System) configuration and customization; and, 3) Reverse Engineering Existing Applications. Each of these scenarios could be the focus of individual papers. BSC is heading in the direction of implementing more and more OTSS applications. Scenario two is the most immediate and problematic. It is the subject chosen for this paper largely because of the nature of commercial off the shelf systems and the complications presented when integrating them with an Enterprise Architecture. Vendors of OTSS applications (e.g. Siebel; AssurX, SAS, etc.) design their products to meet functional needs of a business domain that satisfy the current set of practices for that OTSS's intended use.

3 OTSS Configuration and Customization Overview

OTSS purchases are alternatives to in-house developments and typically requires configuration that is tailored for specific uses. The implementation and use of OTSS has been mandated across many government and business programs, as such products may offer significant savings in procurement, development, and maintenance. Motivations for using OTSS components include hopes for reduction of overall system development and costs (as components can be bought or licensed instead of being developed from scratch) and reduced long-term maintenance costs. In software development, many had considered OTSS to be the silver bullet (to reduce cost/time) during the 1990s, but OTSS development came with many not-so-obvious tradeoffs. Initial cost and development time can definitely be reduced, but often at the expense of an increase in software component-integration work and a dependency on third-party component vendors. In addition, since OTSS software specifications are written externally, government agencies and businesses sometimes fear future changes to the product will not be compatible.[5]

Within BSC there is a significant number of commercial off the shelf systems. Quite often many of the same business subject areas are represented and replicated dissimilarly across disparate OTSS environments. The complications presented when integrating them with an Enterprise Architecture is magnified when attempting to decipher the meaning and semantics of the data and processes implemented across multiple OTSS environments.

4 Enterprise Metadata to Address Integration

To address these inconsistencies between multiple OTSS environments, this paper proposes the use of Object Role Modeling to discover and document the semantics of each business area as depicted in Figure 2 below.

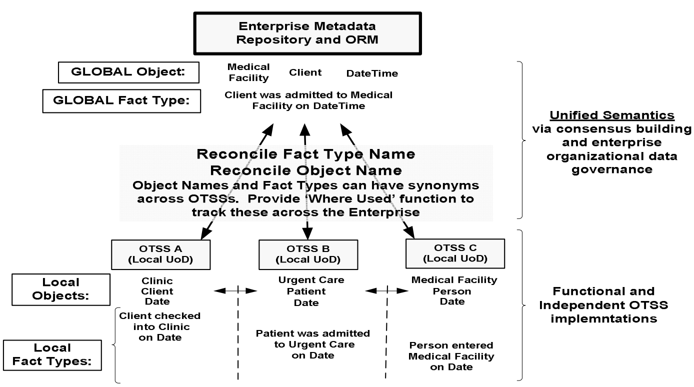


Fig. 2. Unified Metadata Repository[3]

Each OTSS vendor has its own independent representation of the same idea. The enterprise level ORM, represented in Figure 2 above as Global Object and Global Fact Type, is the consensus among multiple business functions in BSC involving disparate OTSSs. The purpose of the Enterprise Metadata Repository and ORM is to harmonize the semantics, objects, and fact types. Synonyms are kept and tracked across all OTSS implementations.

5 OTSS Integration Process

Figure 3 below presents the process flow for a conceptual data model and its transformation to an implemented system and the return of metadata back to the enterprise metadata repository for existing or new OTSS applications. A detailed description of OTSS integration process follows in the subsequent section.

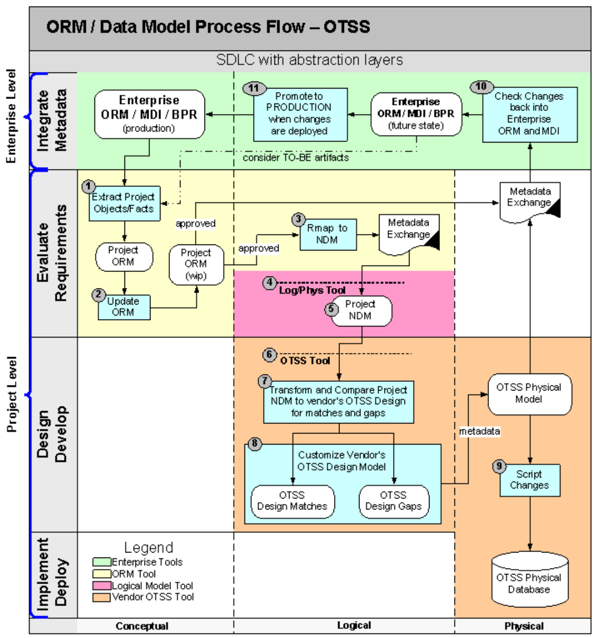


Fig. 3. ORM / Data Model Process Flow – OTSS

5.1 Detailed Description of OTSS Integration Process

Each number in Figure 3 is explained in this section:

(1) Extract Project Objects/Facts

The Enterprise Architect and/or Project Data Architect scope a business domain (sub-schema) and extract a subset of applicable ORM fact types and applicable constraints from the Enterprise ORM for the project based on requirements. A major characteristic of the Enterprise ORM is the generic-ized capture of fact types and constraints. Business domain constraints are typically stronger than their Enterprise ORM counterparts. The architect and/or modeler must also consider pending changes that may exist in the "future state" Enterprise ORM. Together (production and/or future state), the extracted objects are flagged in the Enterprise ORM with an 'Intent'; either "Investigate" or "Modify". Other check-out information is supplied to assist in tracking the changes with other project artifacts.

(2) Update ORM

The Project Data Architect and/or Enterprise Architect meet with Business Stakeholders/Subject Matter Experts (SME's) to apply enhancements and new requirements as illustrated in Figure 4 below. At the end of these modifications the project data model is peer reviewed and approved. The approved Project ORM changes are reconciled and checked back into the Enterprise ORM "future state" model via the Metadata Exchange. These pending changes are not placed back into production until the entire project has been deployed into production.

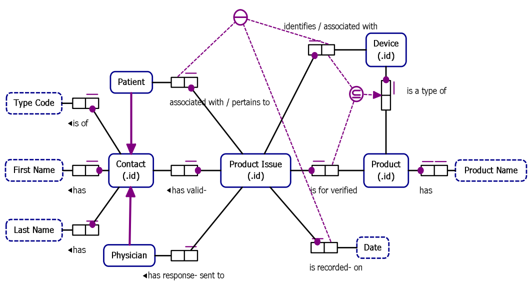


Fig. 4. Project ORM – Updated Subset

(3) Rmap to NDM

The approved project ORM is mapped into the Neutral Data Model (NDM) as a reference model. The Rmap (relational mapper) procedure in ORM is used to map the ORM into a normalized relational data model. All artifacts from the ORM model are transferred to the reference model, these include: notes, descriptions, constraints, rules, tables, columns, data types, etc.

(4) Transition into the Logical/Physical Modeling Tool

The information that defines the requirements (metadata) is formatted and transferred as an input to the Logical Data Modeling tool. The exchange must supply the ORM Fact Types and ORM Objects to the Logical Modeling tool in the form of extended properties to the Logical Model objects. In the OTSS environment, physical data modeling is normally done by the OTSS design tool, although the modeler does have the option of doing this physical model if required, refer to step (9).

(5) Project NDM

The NDM is used to verify that all ORM defined structures, constraints, and business rules are adhered to and accounted for in the implementation of the OTSS. The NDM is rarely, if ever, used to implement directly into a physical model. Almost always, the NDM is transformed into an OTSS design model. The Data Architect has the option of transforming the NDM into an equivalent data model that can assist in transferring the NDM business requirements to the OTSS in a more straight forward manner. Figure 5 represents the mapping of Figure 4 to a Neutral Data Model.

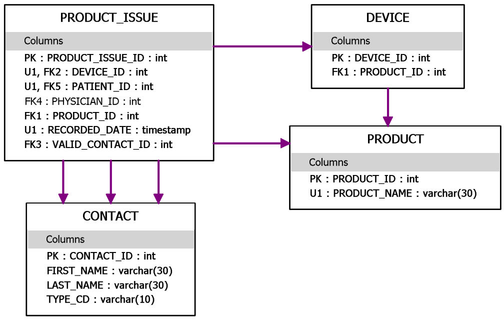


Fig. 5. Project Neutral Data Model

(6) Transition into the OTSS Design Tool

The NDM information that defines the requirements (metadata) is formatted and transferred as an input to the OTSS Design Tool. Each OTSS has a vendor specific set of configuration and customization interfaces that allow the designer to map the business specific requirements to the OTSS application.

(7) NDM Transform and Compare

The Data Architect and the OTSS Design Analyst transform and compare the Project NDM to vendor's OTSS Design for matches and gaps. The original ORM Fact Types and ORM Objects must be accounted for in the OTSS mapping for extraction in follow-on steps, refer to step (10).The blue and red lines in Figure 6 represent the mapping between the NDM data structures and the OTSS data structures for the project.

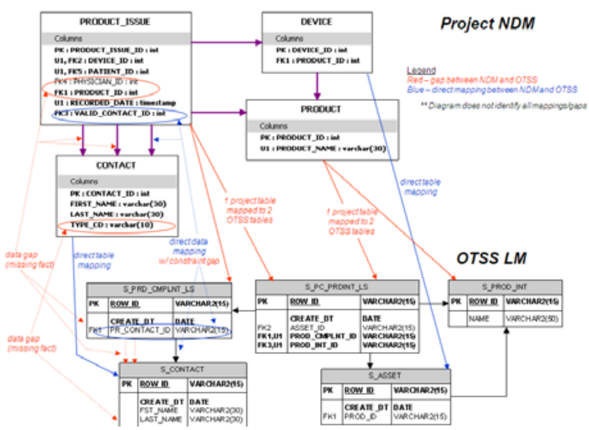


Fig. 6. Project NDM vs. OTSS Logical Model

There are many other gaps, not represented in the above example. It is imperative that all data gaps must be addressed and coded into the OTSS target tables together with their corresponding constraints and rules.

(8) OTSS Configure and Customization

Mappings to existing OTSS structures and identified gaps are documented in Figure 7 below. The following table details the traceability of mapping the ORM Fact Types to their NDM data structures and then on to the OTSS physical data structures. This metadata will be sent back to the Enterprise ORM / MDI / BPR repository.

Key: Red - gap between NDM and OTSS; ☐ - Example: 1 NDM table mapped to 2 OTSS tables
Blue - direct mapping between NDM and OTSS;

Fact ID	Fact	NDM Table	NDM Column	OTSS Table	OTSS Column
F1	Product Issue is identified by ID	PRODUCT_ISSUE	PRODUCT_ISSUE_ID	S_PRD_CMPLNT_LS	ROW_ID
F2	Product is identified by ID	PRODUCT	PRODUCT_ID	S_PROD_INT	ROW_ID
F3	Product Issue is for verified Product	PRODUCT_ISSUE	PRODUCT_ID	S_PC_PRDINT_LS	ROW_ID
F1*	Product Issue is identified by ID	PRODUCT_ISSUE	PRODUCT_ISSUE_ID	S_PC_PRDINT_LS	PROD_CMPLNT_ID
F2*	Product is identified by ID	PRODUCT	PRODUCT_ID	S_PC_PRDINT_LS	PROD_INT_ID
F4	Product is identified by Product Name	PRODUCT	PRODUCT_NAME	S_PROD_INT	NAME
F6	Product Issue has valid Contact	PRODUCT_ISSUE	CONTACT_ID	S_PRD_CMPLNT_LS	PR_CONTACT_ID
F7	Product Issue pertains to Patient	PRODUCT_ISSUE	PATIENT_ID	S_PRD_CMPLNT_LS	X_PATIENT_CONTACT_ID
F8	Product Issue has sent response to Physician	PRODUCT_ISSUE	PHYSICIAN_ID	S_PRD_CMPLNT_LS	X_PHYS_LET_CONTACT_D

Fig. 7. Project to OTSS Mapping

The mappings are used in the OTSS design interface for configuring the OTSS. The remaining gaps are customized in the OTSS using either the vendor's interface (see example in Figure 8) for extending the functionality of their product or a third party platform that can be utilized to effectively and efficiently extend the OTSS application.

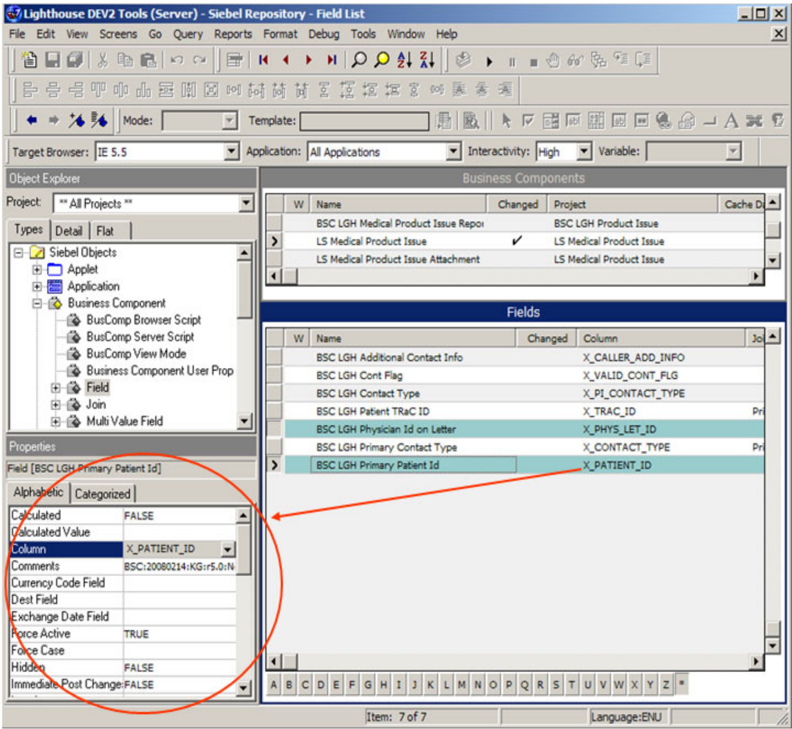


Fig. 8. OTSS Customization Interface

(9) OTSS Script Changes

The OTSS Design model creates scripts for physical database enhancements. Any gap extensions also need to be added to the set of change scripts and builds. Generated and/or manually created SQL ALTER scripts are applied.

(10) Reconcile and Check-In to the Future State

Metadata for OTSS mappings and Add-Ons are extracted for Metadata Integration (MDI). This is used to support traceability and 'where used' functions. Steps (7) and (8) must supply the ORM Fact Types and Objects to the OTSS design model so that they may be extracted for checking back into the Enterprise ORM and MDI future state model. For example, fact types and constraints in the business domain (sub-schema) model are promoted back into the Enterprise ORM. The Enterprise ORM tracks differences between business domains that have differing constraints on population but have the exact same fact type (semantics).

(11) Promote to Production

Metadata for OTSS maps and Add-Ons are promoted back into the Enterprise ORM / MDI / Business Process Requirements (BPR) production model.

6 Future Considerations

Putting the architecture in place to make it possible to leverage ORM and the associated metadata has a far more reaching goal in mind. The evolution of database management systems beyond the traditional relational database management systems that have been around for the last 40 years is now accelerating resulting in more advanced programs capable of managing data of all kinds (e.g. structured and unstructured, marked-up, columnar database, etc.) Many businesses are already implementing Knowledge Centers that evaluate and make available external data sources to internal database management systems in an enterprise.

A recent MarketWatch article by Charles Silver comments on this phenomenom[6]: "Spreading across an increasingly diverse array of data centers, computing will inevitably become much more distributed than it currently is, bringing with it new data-management, architectural and performance challenges. For instance, querying distributed data sources is a new issue that businesses will encounter as cloud computing grows in popularity. A database management system that can query data across multiple distributed databases located in multiple geographically dispersed data centers, including cloud data centers, is what we define as a cloud database management system."

All data reconciliation requires some fundamental understanding and agreement on the semantics of each source of data.

7 Conclusion

Many of the components required for this process have been put in place and validated with two projects: Siebel CRM and AssurX (CATSWeb). In a project that is currently in progress, BSC is applying this process to a set of reports in SAS Analytics. Currently BSC is dependent on nORMa and VSEA for the semantic data requirements model management.

A weakness of nORMa and VSEA is the lack of a repository capability of maintaining logical and physical models. Consequently, an evaluation of technology is currently underway to select a more robust repository platform specifically for logical and physical data model transformations and metadata management. As part of the larger Enterprise Architecture set of domains, the business process applications infrastructure models will be fully integrated in terms of traceability and their relationship to underlying data structures which is captured very well with ORM. The relationship from ORM to the vendor OTSS and on to the Enterprise Business Glossary[4] gives ORM a portal into the other connections that are available via the Business Glossary, e.g.

- where the business term is used in OTSS applications
- how the semantics for various usages of the term is mapped to the OTSS
- facilitates a direct connection to the ORM Object and the OTSS physical database to assist in profiling the production data that is directly tied to these via a metadata manager across multiple OTSS environments

BSC is continuing to develop and evolve an integrated data architecture solution for an environment that builds on the goals and directions of BSC's Enterprise Architecture team.

References

1. Chevalier, J., Pramann, M.: Boston Scientific Corporation, Enterprise Organizational Governance, Enterprise Architecture White Paper (2009)
2. Chevalier, J., Pramann, M.: Boston Scientific Corporation, Enterprise Architecture Value, Enterprise Architecture White Paper (2009)
3. Shelstad, B., Hallock, P., Dela Cruz, N., Barden, D.: Object Role Modeling Enabled Metadata Repository. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 634–646. Springer, Heidelberg (2007)
4. Dela Cruz, N., Schiefelbein, P., Anderson, K., Hallock, P., Barden, D.: ORM and MDM/MMS: Integration in an Enterprise Level Conceptual Data Model. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 457–463. Springer, Heidelberg (2010)
5. Excerpt from Wikipedia, Commercial off-the-shelf (2011)
6. Silver, C.: Why cloud computing must evolve Commentary: Unstructured data are a challenge to current technologies. MarketWatch Commentary (July 11, 2011)

An Advanced Generation Model for a Security Data Warehouse

Baba Piprani[1] and Denise Ernst[2]

[1] MetaGlobal Systems,
Canada
babap@attglobal.net
[2] Canadian Payments Association,
Canada
dernst@cdnpay.ca

Abstract. With all the modernistic web based tools available today, it is ironic that managing a security program in an organization is often relegated to a paper-pencil exercise using outdated information, with an on-going guessing game as to the status and inventory of installed controls, equipment configurations etc. Tracking the incessant onslaught of security breach attempts occurring at an ever increasing pace often is a nightmare. A Fact Based model along with a process model is presented here as a candidate for security information to be contained in a BI-style security Data Warehouse, detailing the primary facts and artifacts of an organization's security program framework and security strategy. The model enables one to draw intelligence from security events, current state of security management and training, risk communication, security architecture and administration controls in place, standards being followed etc., and essentially promotes the concept of availability of security intelligence---data warehouse style.

Keywords: Security, ORM, data warehouse, metamodel, security breach.

1 Current Scenarios and Gaps

With all the modernistic web based tools available today, it is ironic that managing a security program in an organization is often relegated to a paper-pencil exercise using outdated information, with an on-going guessing game as to the installed controls, equipment configurations etc.

Organizations are notorious in playing laggards or being the late majority in introducing security within their environment [1] as can be seen in Fig. 1, sometimes until it is too late---as has been evidenced in current news items with frequent attacks.

Security management personnel are hard pressed to keeping up-to-date on the state of affairs on current security measures and security management controls within their organizations---leave alone tracking the incessant onslaught of security breach attempts that are occurring at an ever increasing pace.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 288–297, 2011.
© Springer-Verlag Berlin Heidelberg 2011

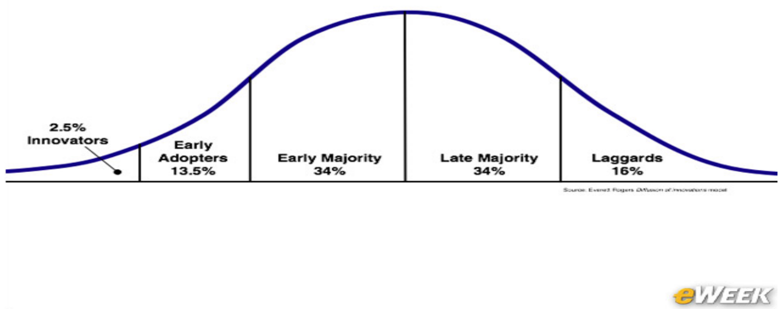


Fig. 1. Security Implementation lifecycle model

The significance of recognizing cyber threats has essentially undergone a paradigm shift in that the US Defense Department considers cyberspace as an operational domain, similar to land, air, sea and space---"Treating cyberspace as a domain means that the military needs to operate and defend its networks, and to organize, train and equip its forces to perform cyber missions"[2]. But do we realize that while we have multitude of apps and data warehouses with data pertaining to financial matters, and things like supply and inventory items for the land, air, sea and space domains, including strategic operational schedules, command and control data etc., hardly anything exists on the inventory of security configuration items (a configuration item is a basic unit of configuration management).

The typical current state of affairs in an organization is to play catch-up on the availability of security intelligence information. Wouldn't it be a godsend to have Security data and intelligence available, Business-Intelligence-style, at your fingertips as in a data warehouse type environment?

Information security practices protect the availability, confidentiality, and integrity of assets for programs and services essential to fulfilling an organization's business obligations. The organization's safeguards are usually commensurate with the potential impact to the organization's finances, operations, and/or reputation if the asset was compromised (sensitivity level).

This paper establishes an advanced generation model for a security data warehouse by first defining a foundational process model that lays the boundaries of the intended candidate scope for Information Security Management. The content and boundaries that are established in this process model were driven by ISO/IEC 27001- Information security management systems family of international standards [3], under the general title *Information technology --- Security techniques.*

Having established the scope and boundaries in the process model, corresponding supporting Fact Based Models are presented here as candidates for security information to be contained in a BI-style security Data Warehouse, detailing the primary facts and artifacts of an organization's security program framework and security strategy. The models enables one to draw intelligence from security events, current state of security management and training, risk communication, security architecture and administration controls in place, standards being followed etc.

The depicted Fact Based Model in this paper is advanced in the sense that this overall model is designed and architected as a generic model that is able to universally address any and all types of items requiring security---borrowing the concepts and terminology for configuration management initially established for a Hardware and Software Configuration Item in US DoD-STD-2167A [4]. The European Space Agency extended the concept of a Configuration Item (CI) to address configuration management of complex products by dividing the complex product into manageable segments and components through is lifecycle in terms of Identification, Control, Recording and follow-up, verifying required attributes, and auditing [5].

An advanced generation CI-based security data warehouse design would be based on a highly normalized attributed data model that would be typically derived from the base concepts portrayed in the Fact Based Models in this paper---of course, by completing the data model by adding implementation artifacts and implementation considerations. It is important to note that the resulting attributed data warehouse data model would be architected on the lines of the design as depicted in the advanced generation data warehouse and data quality firewall [6] and in itself be a CI in terms of implementation of security controls.

The fact based models and process model presented in this paper seek to address a very wide gap that currently exists in organizations with respect to non-automated means of maintaining configuration management information on the organization's inventory of security infrastructure---depicting World War II style scenarios of manual whiteboards and sticky notes with obsolete documentation.

The models presented here promote the concept of availability of security intelligence---data warehouse style---in a manner that is based upon the critical success factors of security information facts, currency, data quality, integrity etc., and addresses a wide gap that exists with the Security personnel today in terms of where-the-heck-do-I-find-this-information. In particular, the collections of facts would enable a connecting-the-dots exercise in intelligence gathering and avoid intelligence failures like the Pearl Harbour intelligence failure in 1941, where the information on the Japanese ships and frigates being in position was available piece-meal and the US had access to the Japanese diplomatic codes, but no-one put it together that the attack was imminent---similarly with the Israeli failure to recognize the imminence of war until the morning of October 6, 1973.

2 The Basic Process Model for Information Security Management

Fig. 2 depicts a process model (IDEF0 decomposition notation [7]) for information security management depicting the typical classifications of functions denoted by P1-P5, as being performed towards adopting a standards-based, or organized approach in information security management.

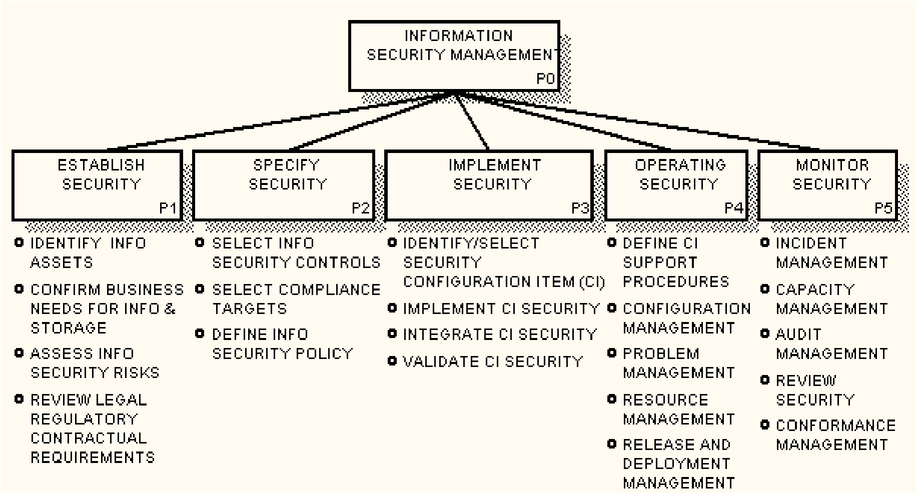


Fig. 2. Basic Process Model for Information Security Management

3 Supporting Fact Based Models

The following sections portray supporting Fact Based Models for each of the relevant process model function grouping shown.

Note that the notation used in the following Fact Based Models are based on the ORM/NIAM schema [8] for Fact Types (rectangle pairs), Object types---Non-Lexical Object Types (solid circles) NOLOTs, Lexical Object Types (dotted circles) LOTs, role total constraint (tick mark on the role line) etc. and that are native to the Ooriane software tool. Thanks to Ooriane Corp. for the use of Ooriane Designer and Ooriane Semantic Analyzer for ORM/NIAM graphics.

3.1 Functional Grouping for P1 ESTABLISH SECURITY

The first Fact Based Model as shown in Fig. 3 addresses P1 ESTABLISH SECURITY functional group. Facts supporting IDENTIFY INFO ASSETS are depicted as being associated with ORG UNIT addressing the supertype of CONFIGURATION ITEM (acronym CI) ---which has subtypes CI DATA, CI SW, CI INTERFACE, CI HW---artifacts essentially based on NIST listing of risk assessment activities [9]---that address the sub-functions ASSESS INFO SECURITY RISKS, and REVIEW LEGAL REGULATORY CONTRACTUAL REQUIREMENTS. The associations between ORG UNIT and CI DATA portray the CONFIRM BUSINESS NEEDS FOR INFO AND STORAGE sub-functions. Table 1 shows the accompanying object type definitions.

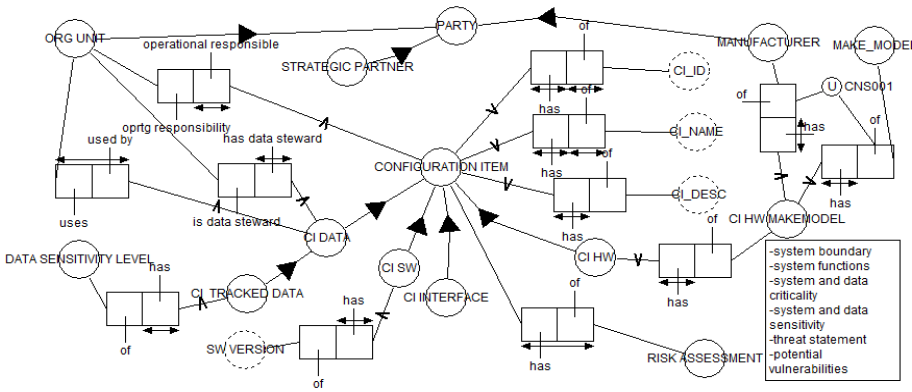


Fig. 3. Fact Based Model for P1 ESTABLISH SECURITY

Table 1. Object Type definitions for P1 ESTABLISH SECURITY

Object Type	Description
CI DATA	A CI of type : data
CI HW	A CI of type : hardware
CI HW MAKEMODEL	A CI hardware's identification based on the manufacturer's identification scheme
CI INTERFACE	A CI of type : interface
CI SW	A CI of type : software
CI TRACKED DATA	A CI of type : data that is to be tracked for audit and security purposes that is subject to any specializations of facts
CONFIGURATION ITEM	An item of value to an organization such as: printer, workstation, laptop, cell phone, smart phone, server, data center, USB drive, token or access card, software and physical media, strategic, policy and planning documents, personal information, financial information, asset inventories, security and risk assessment reports, information on IT network, architecture, IP addresses, system logs, password and account information
DATA SENSITIVITY LEVEL	the information classification level of an information asset such as: unclassified, confidential, protected A, protected B, secret, top secret, for your eyes only
MAKEMODEL	A manufacturer's designation to identify differences within a model grouping by denoting a finer grouping of items with identical or related characteristics within a larger group.
MANUFACTURER	A Party that produces something that is of use within the scope of the organization
ORG UNIT	A subtype of Party that has an administrative and functional structure within the organization
PARTY	A person or a group that is participating in some action or is involved in some transaction with the affairs of the organization
RISK ASSESSMENT	An approach used to identify the risk of an event such as: checklist (e.g. survey), general (e.g. internal assessment), comprehensive (e.g. threat risk assessment)
STRATEGIC PARTNER	An external organization that can provide early warning, assistance, and advanced intelligence, or that the organization needs to communicate with, related to a security event such as: local emergency response teams (e.g. Fire Dept., Police…), local incident management teams (e.g. provincial emergency management…), local crisis management teams (e.g. Red Cross…), government (municipal, provincial, federal), service providers (e.g. security solution providers, financial/payroll, telecommunication…), forensic investigators, telecommunication, special protection groups (e.g. RCMP…), media (public communication)

3.2 Functional Grouping for P2 SPECIFY SECURITY

The functional grouping for P2 SPECIFY SECURITY as shown in Fig. 4 essentially enables an ORG UNIT to DEFINE INFO SECURITY POLICY, and establish pre-defined RELATIONSHIP TYPEs with STRATEGIC PARTNERs. Strategic Partners are those that the org unit needs to immediately confer and inform in case of an attack

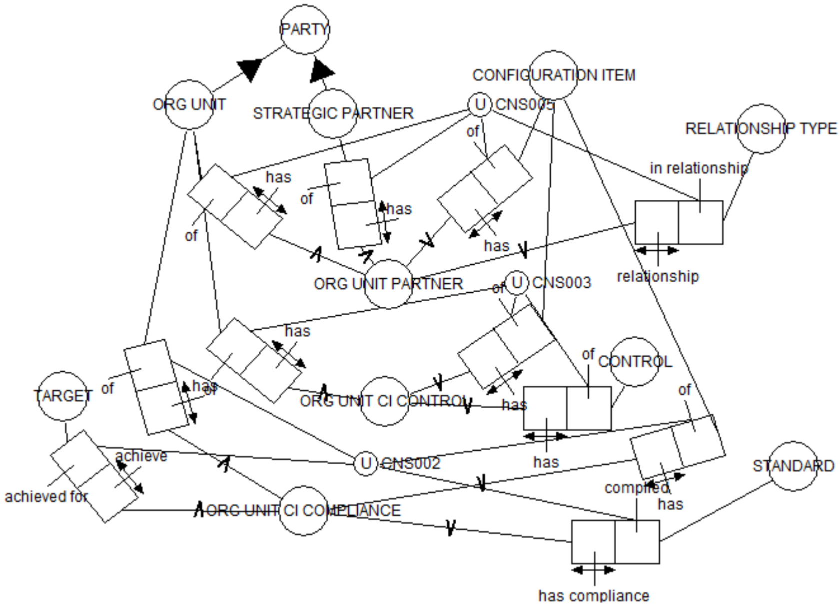


Fig. 4. Fact Based Model for P2 SPECIFY SECURITY

Table 2. Object Type definitions for P2 SPECIFY SECURITY

Object Type	Description
CONTROL	Safeguards that protect assets such as: information security policy, digital authentication standard, information protection standard, new-hire/change/termination process(es), personnel security briefing, firewall, encryption, anti-malware, safes and locks, proximity card reader, authentication and ID tools, cameras and alarms
ORG UNIT CI COMPLIANCE	The facts related to an organizational unit's compliance for the given CIs
ORG UNIT CI CONTROL	The facts related to an organizational unit's controls for the given CIs
ORG UNIT PARTNER	The facts related to an organizational unit's strategic partners for the given CIs
RELATIONSHIP TYPE	The category of relationship with the strategic partner
STANDARD	The standard identifier that the organization adheres to from e.g.: NIST, ISO, CSE (Canadian Security Establishment), X.500 etc.
TARGET	A metric to establish the acceptable threshold for a security event such as: # of occurrences of malware, patch level, failed access attempts, attempts to access blacklisted sites, capacity threshold, acceptable system downtime

or imminent threat. Also shown is whether compliance targets selected by the org unit are projected or achieved based for a given configuration item in ORG UNIT CI COMPLIANCE. The SELECT INFO SECURITY CONTROLS enables the depiction of what controls were selected on which CONFIGURATION ITEM for the ORG UNIT. Table 2 shows the accompanying object type definitions.

3.3 Functional Grouping for P3 IMPLEMENT SECURITY

The functional grouping for P3 IMPLEMENT SECURITY as shown in Fig. 5 records facts on whether compliance targets selected by the org unit are projected or achieved based for a given configuration item in ORG UNIT CI COMPLIANCE. The fact based model shows for example the install of antivirus software, hardware firewall, software firewall, firmware, hardware-to-operating system (OS) compatibility etc., so as to enable an org unit to implement security and conduct configuration management on its security resources, integrate CI security, and validate CI security against the previously established models. Table 3 shows the accompanying object type definitions.

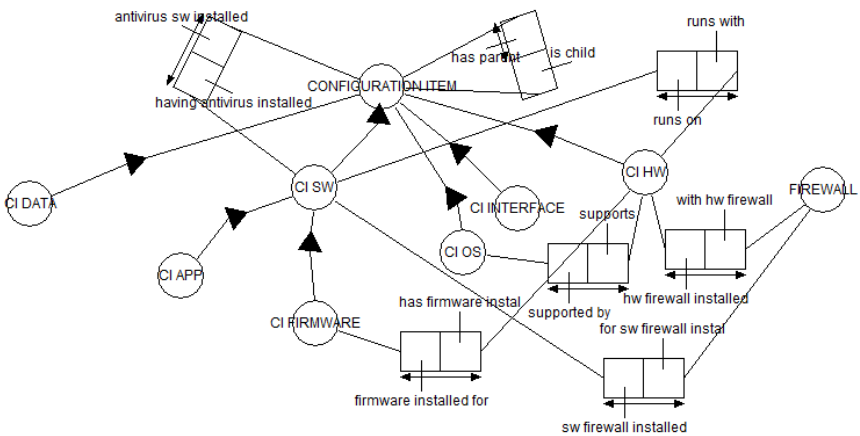


Fig. 5. Fact Based Model for P3 IMPLEMENT SECURITY

Table 3. Object Type definitions for P3 IMPLEMENT SECURITY

Object Type	Description
CI APP	A CI of type : application
CI FIRMWARE	A CI of type : firmware
CI OS	A CI of type : operating system
FIREWALL	A configuration or device set configured to permit or deny network or pass-through transmissions based upon a set of controls

3.4 Functional Grouping for P4 & P5 OPERATING SECURITY, MONITOR SECURITY

The functional grouping for P4 and P5 as shown in Fig. 7 entail security incident operating, monitoring, tracking, problem management, auditing etc. The fact based model in Fig. 5 shows how SECURITY INCIDENTs can be tracked in terms of

CHARACTERISTIC, its SERIOUSNESS LEVEL, the ORG LOCATIONs, the PREVENTIONS-STEP MEASURES, and the exact CONFIGURATION MANAGEMENT-CI-ORG UNIT combination that was involved. Also shown is the state of affairs for INCIDENT CI i.e. the exact CONFIGURATION ITEM involved in the SECURITY INCIDENT as to tracking start, end and unserviceable time. Table 4 shows the accompanying object type definitions.

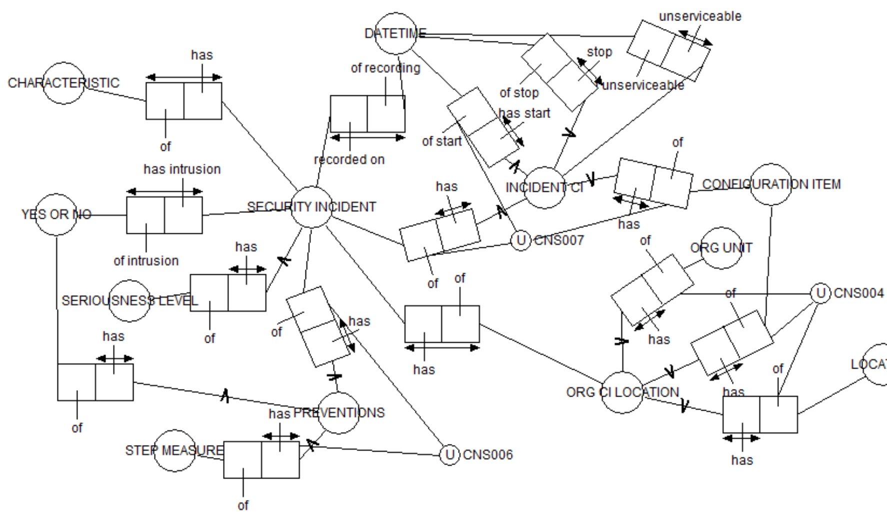


Fig. 6. Fact Based Model for P4 & P5 Operating Security, Monitor Security

Table 4. Object Type definitions for P4 & P5 OPERATING SECURITY, MONITOR SECURITY

Object Type	Description
CHARACTERISTIC	A property of an event that threatens the confidentiality, availability, or integrity of an asset such as: denial of service attack, malicious code, unauthorized scans or probes, intrusions (cyber or physical), insider attacks
DATETIME	Time and date
SECURITY INCIDENT	An adverse event that has resulted in a loss of confidentiality, availability or integrity of an asset such as: malware, theft, cyber breach, data breach, defacement of a corporate website
INCIDENT CI	A given CI that is involved in an adverse event
LOCATION	A spatial area designated for purposes of security
ORG UNIT	A designated division or functional unit of an organization
PREVENTIONS	Control used to prevent a security incident such as: authentication and ID tools, firewall, anti-malware software, policy, encryption mechanisms, safes and locks, proximity card reader, data loss prevention tools
SERIOUSNESS LEVEL	The degree of impact to the organization if an asset was compromised such as: low, moderate, high
STEP MEASURE	control used to monitor changes in the operational environment such as: log aggregation and correlation tools, cameras, exception registry, change management
YES OR NO	Yes or no indicator

4 Conclusion

With Cyber threat incidents occurring at an increasing pace, the resource management infrastructure and framework for managing the various security components does not appear to be as organized as a financial or Customer Relationship Management data warehouse---and in fact is generally circa pre-automated and primitive state. The advanced generation model of a candidate security data warehouse shown in this paper addresses this wide gap and depicts an organized collection of designs extracted from ISO security standards, NIST security standards and guidelines. The model has also been through a thoroughness validation exercise by mapping various other industry sources, whitepapers etc. The models presented here promote the concept of availability of security intelligence---data warehouse style---in a manner that is based upon the critical success factors of security information facts, currency, data quality, integrity etc. The models enable one to address a wide gap that exists with the Security personnel today in terms of where-the-heck-do-I-find-this-information, and more importantly, enable one to connect-the-dots.

The views expressed in this paper are those of the authors. No responsibility or liability for them should be attributed to the Canadian Payments Association.

References

1. Cloud Computing: Cloud Computing Security: 10 Ways to Enforce It, eWeek.com, Ziff Davis Enterprise Holdings, Inc., July 06 (2011), http://www.eweek.com/c/a/Cloud-Computing/Cloud-Computing-Security-10-Ways-to-Enforce-It-292589/?kc=EWKNLINF07132011STR1
2. Ottawa Citizen July 15 2011 Section A11, page 3, Foreign hackers hit Pentagon supplier, http://www.ottawacitizen.com/technology/Foreign+hackers+Pentagon+supplier/5105974/story.html
3. ISO/IEC 27000:2009, Information security management systems family of standards consisting of ISO/IEC 27001:2005 – Requirements, ISO/IEC 27002:2005 – Code of practice for information security management, ISO/IEC 27003 – Information security management system implementation guidance, ISO/IEC 27004:2005 – Measurement, ISO/IEC 27005:2008 – Information security risk management, ISO/IEC 27005:2008 – Information security risk management, ISO/IEC 27006:2007 – Requirements for bodies providing audit and certification of information security management systems, ISO/IEC 27007 – Guidelines for information security management systems auditing, and ISO/IEC 27011 – Information security management guidelines for telecommunications organizations based on ISO/IEC 27002
4. DOD-STD-2167A, Military Standard: Defense System Software Development, US Department of Defense (February 1988) [S/S by MIL-STD-498]
5. ESA training for small and medium sized enterprises, Configuration Management training materials. European Space Agency (February 2008), http://www.esa.int/esaMI/Events_Training/SEMNNWK8IOE_0.html
6. Piprani, B.: Using ORM-based Models as a Foundation for a Data Quality Firewall in an Advanced Generation Data Warehouse. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1148–1159. Springer, Heidelberg (2006)

7. Integration Definition for Function Modeling (IDEF0), Federal Information Processing Standard 183, National Institute of Standards and Technology (NIST) (December 1993)
8. Nijssen, G.M., Halpin, T.A.: Conceptual Schema and Relational Database Design. Prentice Hall, Victoria (1989)
9. Stoneburner, G., Goguen, A., Feringa, A.: Risk Management Guide for Information Technology Systems, National Institute of Standards and Technology, NIST Special Publication 800-30 (July 2002)

Modeling Slowly Changing Dimensions in ORM

Ron McFadyen

University of Winnipeg, Winnipeg, Manitoba
r.mcfadyen@uwinnipeg.ca

Abstract. A data warehouse is a subject oriented, integrated, non-volatile, and time variant collection of data in support of management's decisions. Dimensional modeling is a design technique used extensively in the design and construction of data warehouses. To account for different requirements there are several standard variations designers may utilize in their dimensional designs such as outriggers, bridge tables, mini-dimensions, slowly changing dimension techniques, etc. Our focus in this paper is on the representation of the dimensional schema as an ORM schema where slowly changing dimension techniques Types 1 and 2 are used. We examine the nature of the ORM schema - its objects, roles, constraints, rules and transactions.

Keywords: object-role model, dimensional model, data warehouse, slowly changing dimension.

1 Introduction

A data warehouse is a subject oriented, integrated, non-volatile, and time variant collection of data in support of management's decisions [1]. A data warehouse is organized by subject areas such as customer sales, inventory, etc. This organizational property facilitates the work of business analysts who need to understand performance of business processes. Data targeted to the data warehouse may come from many sources and therefore may need to be integrated. Non-volatile means the data is generally not altered once it arrives in the warehouse. The warehouse is a repository for historical data describing events of the enterprise over time. A data warehousing environment involves data sources from which data is obtained and propagated to a data warehouse via extract, transform, and load (ETL) processes.

Dimensional modeling (DM) [2] is a design technique used extensively in the design and construction of data warehouses. At its simplest, a dimensional model implemented as a relational database has one table designated as the *fact* table which participates in many-to-one relationships with other tables known as *dimension* tables. The term *fact* is used differently in DM than how *fact* is used in an object-role modeling (ORM) [3] context. Where necessary, we will differentiate in usage through the terms *fact table* (DM) and *fact type* (ORM). A fact table is characterized as containing attributes that are measures of a business process (e.g. quantity, dollar amount, etc. for sales), and the dimension tables are characterized as containing attributes that describe measures (an item's category, the date of sale, etc.). In general these dimensional designs are considered to be much simpler for business analysts to work with compared to normalized designs that exist in source systems.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 298–308, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Several texts including [2], [4-6] describe the basic dimensional model as having a single fact table related via mandatory many-to-one relationships with dimension tables (no foreign key in the fact table can be null). Dimension tables have a primary key (PK) that is a surrogate key (SK); other approaches involving composite keys (natural key is used in conjunction with some date and or time related attribute) are purposely avoided.

To account for different requirements there are several standard techniques designers utilize in their dimensional designs such as outriggers, bridge tables, mini-dimensions, slowly changing dimension techniques, etc. Our focus in this paper is on object-role models for dimensional designs with slowly changing dimensions (Types 1 and 2) and where all attributes except the natural key are subject to change in the source systems. We examine the nature of the ORM schema - its objects, roles, constraints, rules and transactions. As a running example we use the example from [3, Ch. 16].

Many articles such as [7-11] focus on an entity-relationship or UML approach to designing a warehouse. A complete treatment of ORM is found in [3]; several publications such as [12-15] relate to ORM-based warehousing. [12] proposes a fact-oriented approach to multidimensional modeling based on ORM. Specializations to object types are proposed to represent events, dimensions, and hierarchies, and modifications are proposed to the conceptual schema design procedure which is used to derive a warehouse model from user requirements. [3, Ch. 16] presents a user-guided process to generate a relational schema based on an ORM schema. This is done by annotating the model with key-object types (similar to events and dimensions in [12]) and thereby allowing a different physical database, compared to one obtained using the standard RMap procedure [3, Ch. 11], to be generated. [16] proposes ORM constraints that can assist in more options for a physical database.

When we model a data warehouse we are naturally concerned with time due to the temporal nature of the data warehouse. [17] provides a conceptual framework for classifying temporal information. The period of time that certain data is valid must be known or derivable. For each type of data we must determine the temporal granularity. Adding data to a warehouse implies a transaction [18] that alters the database state and we must ensure the database moves from one correct state to another through the use of static and dynamic rules [19-20].

The rest of this paper is organized as follows. We begin in Section 2 with a review of dimensional modeling, Type 1 and Type 2 dimensions. Section 3 presents features of an ORM dimensional schema, its constraints, a transaction model, and database rules to ensure data integrity. Lastly, we summarize our results, and consider future directions.

2 Dimensional Modeling and History in Dimensions

The relational schema in [3, Figure 16.3] provides for recording the history of all sales but there is no provision to track historical data in dimensions. Figure 1 shows the dimensional relational schema from [3] with the changes: the primary keys of dimensions are surrogate keys (note the suffix Sk used for these attributes), Store is a Type 1 dimension, and Item is a Type 2 dimension.

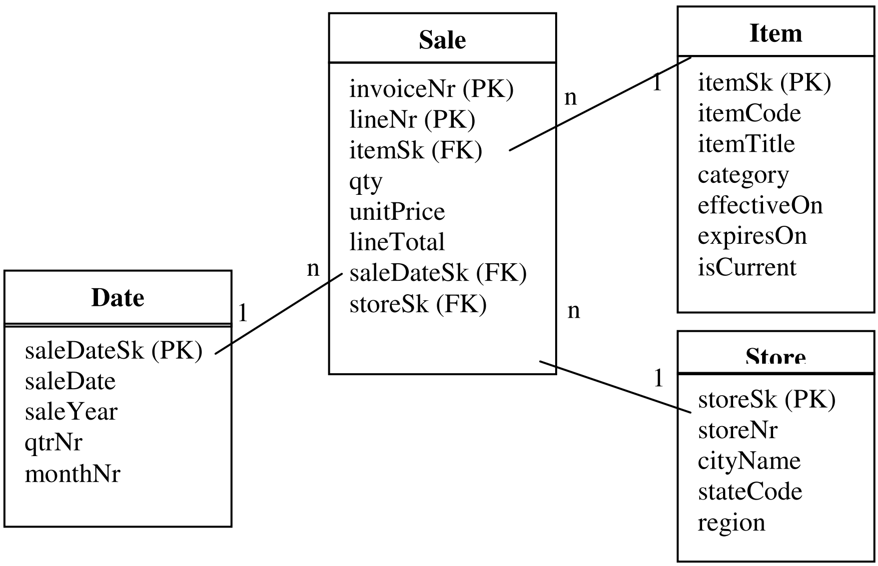


Fig. 1. Relational schema with surrogate keys and Type 1 & 2 dimensions

The consistent use of surrogate keys for all dimensions (including the Date dimension) has some useful implications. SQL joins have a certain symmetry and simplicity because all natural joins become single attribute joins. In addition surrogate keys facilitate additional rows in a dimension table providing for special cases (such as an *unknown* row or a *not yet* row) which reduce the need for queries involving the keyword *null*. Suppose our example were extended to include the ship date for an invoice. Now we would have another many-to-one relationship (ship date) between the fact table and the Date table. Then any fact table rows relating to an invoice that has not shipped would have a foreign key referencing the *not yet* row in the Date dimension. A query to retrieve information relating to invoices that have not shipped would have a filter *not yet* for the ship date.

In situations where history in a dimension is not required a Type 1 dimension is used. The primary purpose for a Type 1 dimension is for storing the most recent information for a dimension entity. When a pertinent change is captured and propagated to the warehouse, the existing row for the entity involved is overwritten.

Surrogate keys allow for more than one row per natural key; this is used to facilitate the historical record for dimensional data. The Type 2 technique requires a new dimension row to be created whenever there is a pertinent change in the source database. The standard approach [2, p. 193] to implement this technique includes *effective date*, *expiry date*, and *current indicator* attributes. When a change occurs in the operational system, the ETL system will propagate that change to the warehouse and insert a new row in the dimension table and modify the previously current row. Recommendations in [2] include that dates always have a value; so if data has not yet expired (it is current), then high values such as Dec 31, 9999 are used instead of null. The motivation for this is to simplify queries: if current information is wanted then filter using the current indicator attribute, or if information related to a specific date is wanted then the SQL BETWEEN operator can always be used.

Type 3 is used in cases where limited history is wanted such as prior and current values. Type 1 and Type 2 are considered the primary workhorse techniques for responding to changes in a dimension. These techniques are provided for in various commercial products as Microsoft's Integration Services [21], Oracle's Warehouse Builder [22], and Embarcadero's ER/Studio [23].

Figure 1 presents the dimensional relational model for our example (originally from [3]) where: dimensions have surrogate keys, Store is a Type 1 dimension, and Item is a Type 2 dimension. A distinguishing characteristic of DM is the star-like appearance of these designs where the fact table is shown at the center.

Type 1 Example. Suppose Store is a Type 1 dimension with relational schema {storeSk, storeNr, cityName, stateCode, region}. Consider a store '66' where the initial record was inserted into the warehouse on Jan 1, 2000 where the region is NorthWest. Suppose on Jan 1, 2011 the store is reassigned to a region named Pacific. Sometime after Jan 1, 2011 the row for store 66 is overwritten, reflecting the change. Type 1 is a simple overwrite of the row with new data, history is lost, and at any point in time only the most recent values are available – see Table 1.

Table 1. Row overwritten in Store (Type 1)

Date written/overwritten	storeSk	storeNr	cityName	stateCode	region
Jan 1, 2000 →	25	66	Seattle	Wa	NorthWest
Jan 1, 2011 →	25	66	Seattle	Wa	Pacific

Type 2 Example. Now suppose Item is a Type 2 dimension with relational schema {itemSk, itemCode, itemTitle, category, effectiveOn, expiresOn, isCurrent}. Consider an item '27' and: suppose the initial record was inserted into the warehouse on Jan 1, 2000 with title Vegetarian Noodles and category Pasta; on Jan 1, 2011 its title is changed to Vegan Noodles; on March 1, 2011 its category is changed to Noodle. ETL processes will result in three records for item 27 with three different surrogate key values – see Table 2. Note that isCurrent is a boolean data type, all dates have a value, and the use of high values for an expiry date. Effective and expiry dates always have a value; if some row has not expired then the expiry date is set to high values in order to avoid queries with nulls.

Table 2. Rows in Item (Type 2)

itemSK	itemCode	itemTitle	category	effectiveOn	expiresOn	isCurrent
11	27	Vegetarian Noodles	Pasta	1/1/2000	12/31/2010	False
…						
60	27	Vegan Noodles	Pasta	1/1/2011	2/28/2011	False
…						
77	27	Vegan Noodles	Noodle	3/1/2011	12/31/9999	True
…						

3 ORM Schemas

We now consider the ORM schema an end user could be working with. Figure 2 shows a result of reverse engineering Figure 1 as an ORM schema (without the details of dimensions at this time). We note the following regarding the reengineered schema:

- The external uniqueness constraint (UC) is derived directly from Sale's primary key.
- Fact types involving Sale and a dimension object type are mandatory for Sale.
- For each binary fact type Sale has a role in, the other role is either played by a measurement value type (e.g. quantity), or by an object type with a surrogate key reference scheme.
- Sale could be shown in a nested form where a binary fact type involving invoice number and line number is objectified, but we prefer this co-referenced form as it illustrates a symmetric design – Sale is shown as an object type at the center similar to the star-like designs of DM.

A Type 1 dimension table has a surrogate key serving as the primary key; Figure 3 shows the ORM subschema for Store as Type 1. Note the following points of interest:

- All roles for the SK object type (except regarding Sale) must have a UC.
- There is a UC on the role played by the natural key in the fact type *has natural key*.
- The fact type *has natural key* is only mandatory for the SK object type.

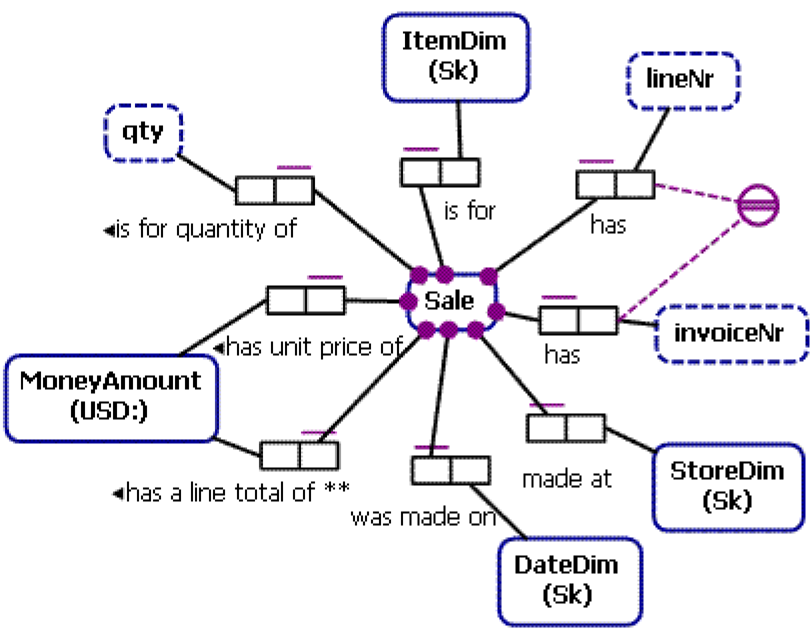


Fig. 2. An ORM schema corresponding to relational schema in Figure 1 (dimension details omitted)

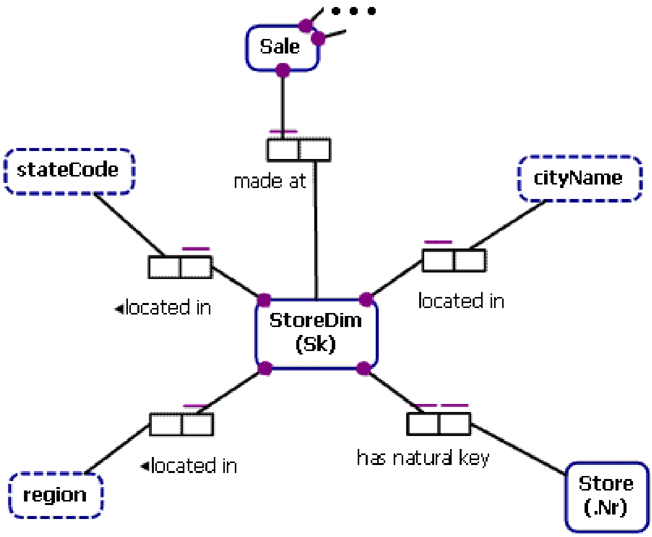


Fig. 3. Store as a Type 1

Now, we consider Type 2 dimensions; a Type 2 dimension table has additional attributes and so a corresponding ORM schema has new object types, fact types, and constraints – see Figure 4.

Note the following points of interest:

- All roles for the SK object type (except regarding Sale) must have a UC.
- Each SK object type must have a time period defined by an *effective on* and *expires on* dates where the periods for one SK object type instance: do not overlap, are complete, and only the last period can represent current data.
- There can be many alternative models as illustrated in [17] where several equivalent conceptual models are presented and referred to as nested, co-referenced, and flattened forms. In this situation of a Type 2 we believe Figure 4 represents the most natural expression since dimensions generally have many descriptive attributes and because there must be an expiry date. This design also continues the appearance of star-like designs – in the case of dimensions the SK object type will be the center.
- The *expires on* fact type is semi-derivable. The periods that comprise the historical record for a natural key are non-overlapping and contiguous forming a sequence of time periods. For all but the last period the expiry date is derived as the next start date minus one unit of time (typically a day, but the granularity depends on the application). The expiry date is semi-derived because the last expiry date is one of two possibilities: it is either high values or a value determined by ETL processes. A similar case applies if we were to consider the *effective on* date because ETL processes determine the first *effective on* date and the others can be derived from the previous *expires on* date. In a later section we give transaction definitions that derive the *expires on* date but which allow the ETL processes to set the last *expires on* date.

- An important part of the Type 2 slowly changing dimension technique is the *isCurrent* attribute specifying if current data is represented or not. In a dimension table only one out of many rows for a specific natural key can represent current values. Modeling examples in [16] are for situations similar to this: (1) there is one politician who is the government head out of many who serve in government, and (2) there is one main city that is the capital out of many main cities. Because of a problem with the schemas generated by RMap, [16] introduces the *unique-where-true* constraint to facilitate alternative schemas. However, this difficulty does not arise here because the *is current* fact type is derivable: for a Type 2 schema *is current* is true for one specific value (high values) of *expires on*. This situation is then a special case obviating the need for *unique-where-true*.

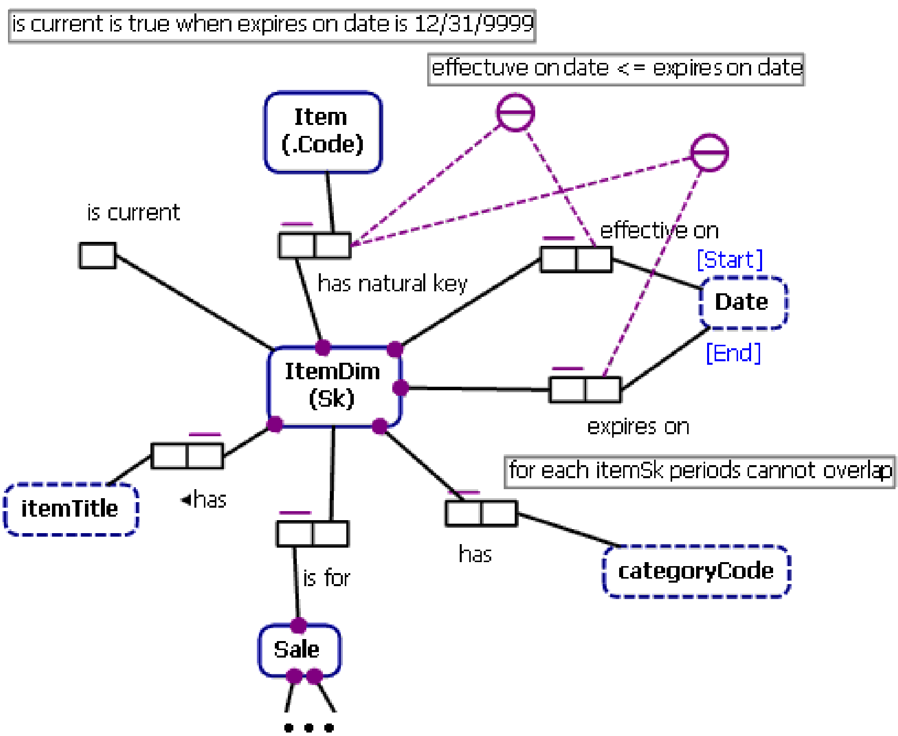


Fig. 4. ORM subschema for Item as a Type 2 dimension

3.1 Transactions and Rules

The 100% Principle [24] states "all relevant general static and dynamic aspects … of the universe of discourse should be described in the conceptual schema". In keeping with this principle we now consider the nature of transactions and rules. Type 1 dimensions are simpler than Type 2 and in the interest of space we consider only Type 2 dimensions; consideration of Type 1 dimensions is left to the reader.

Transactions. Corresponding to transactions (inserts, updates and deletes) in source systems, ETL processes must be able to perform corresponding transactions for the warehouse. In practice there is a need to handle corrections, but we omit those considerations here. In the interest of space we only provide transaction definitions for updates and deletions concerning Type 2 dimensions. Note that these transactions determine the value for the semi-derivable fact type *expires on*.

As an example, consider Table 2 and the last change for item 27 where its category changes from Pasta to Noodle. Using the transaction formalism from [18], we illustrate a specific compound transaction that changes the contents of the warehouse: *execute Apply_Change (77, 27, '3/1/2011','Vegan Noodles', 'Pasta').*

```
Apply_Change (newSK, natKey, startDate, title, category)
Begin
  -- is current is derived and not set here
-- Change the 'previously new' time period
CHANGE IN ItemDim expires on Date
SET Date = startDate-1
WHEN
     ItemDim
       |__ has natural key natKey
       |__ End = HIGH_Values
-- Add the 'new'
  ADD TO ItemDim Values (newSK)
  -- Add the relationships for a type 2 dimension
  ADD TO ItemDim effective on Date VALUES (newSK, startDate)
  ADD TO ItemDim expires on Date VALUES (newSK, HIGH_VALUES)
  ADD TO ItemDim has natural key Item VALUES (newSK, natKey)
  -- Statements for descriptive object types of dimension:
  ADD TO ItemDim has ItemTitle VALUES (newSK, title)
  ADD TO ItemDim has ItemCategory VALUES (newSK, category)
End
```

Now suppose an item is deleted in the source system. Suppose in the source system item 27 (natural key) is deleted on Oct 1, 2011. This requires the corresponding historical record to be ended in the warehouse. Consider the compound transaction:

```
Apply_Delete (27, '10/1/2011'):
  Apply_Delete (natKey, endDate)
  Begin
  CHANGE IN ItemDim expires on Date
  SET Date = endDate
  WHEN ItemDim
         |__ has natural key Item = natKey
         |__ effective on Date = HIGH_VALUES
  End
```

Note that much of these transactions vary only by the name of the dimension and so transaction templates could be used to standardize their formulation.

Rules. Transactions like the above change the state of the database, in particular the state of affairs associated with a particular item. To ensure integrity of historical information a combination of static and dynamic rules [19-20] can be used. Static rules apply to a single state and are encoded in the ORM schema via constraints, fact types, and object types, whereas a dynamic rule constrains the database as it moves from one state to another. As stated in [2], Type 2 dimensions create a perfect partitioning of history where there are no gaps in the historical record.

We propose two rules: (1) to ensure modifications are applied only to the last historical entry representing current information, (2) to ensure no gaps exist when history is extended. An ORM tool could be modified to automatically generate such rules for a Type 2 dimension.

Our first rule ensures only the last historical entry is modified:

```
Context: End
    For each instance modified
          new End <> HIGH_VALUES and
          old End =  HIGH_VALUES
```

Our second rule ensures that a newly added historical entry is the last entry in time period sequence and that there is no gap between it and the previous last entry. To simplify this rule, we utilize the *previous* function [19].

```
Context: Item
    For each ItemDim added
          previous ItemDim.End = Start – 1
          End = HIGH_VALUES
```

4 Conclusion

Dimensional modeling is a constrained form of entity relationship modeling and so it does not have the expressive power of object-role modeling. We have presented an ORM schema for a dimensional model that incorporates Type 1 and Type 2 dimensions. In its basic form a dimensional schema is quite simple and predictable: that part of the ORM schema relating to a fact table can be centered in the layout which radiates outward in a star-like fashion. Due to the attribute-free nature of an ORM schema, the ORM subschemas for dimensions continue the star-like appearance.

A Type 2 dimension must represent complete historical records for each natural key. At most one period (the last one) of a historical record represents current information. The need for a subset constraint and a unique-where-true constraint is obviated because the *is current* fact type is derivable for a Type 2 dimension.

The history for a natural key is represented using time periods with a start and end date forming contiguous periods (no overlaps, no gaps). The *expires on* fact type is semi-derivable and so transactions can be formalized to set the expiry date appropriately.

In keeping with the 100% Principle, we provide transaction specifications and dynamic rules to ensure integrity of data in an ORM dimensional schema. However, this framework might only be utilized in a development environment and not a production environment due to the volumes of data and because of the periodic nature of warehouse updates; it is typical in a data warehouse environment that ETL processes are designed to ensure data integrity.

Type 1 and 2 dimensions represent common modeling cases for a warehouse and could be provided for in an ORM tool. By selecting some pertinent property, specific attributes, transactions, and rules could be automatically added to a schema. This would facilitate the modeling effort, and such a capability may be useful in other situations where default fact types, transactions, and/or rules can be specified.

In this paper we considered the nature of the ORM schema that is equivalent to some relational dimensional schema. That is, we have worked from a relational model reengineering the ORM schema. Another approach to consider is to begin with source system ORM schemas and generate dimensional schemas involving different dimension types. There are a number of other aspects of DM that we have left for future work: keyword dimensions, hierarchical relationships, bridge tables, aggregation, additivity, mixing of Type 1, 2, and 3 in a single dimension, etc.

References

1. Inmon, W.: Building the data warehouse. QED Technical Pub. Group (1992)
2. Kimball, R., Ross, M., Becker, B., Mundy, J., Thornthwaite, W.: The Kimball Group Reader. John Wiley & Sons (2010)
3. Halpin, T., Morgan, T.: Information Modeling and Relational Databases. Morgan Kaufmann Publishers (2008)
4. Kimball, R., Reeves, L., Ross, M., Thornthwaite, W.: The data warehouse lifecycle toolkit. John Wiley & Sons (1998)
5. Adamson, C., Venerable, M.: Data warehouse design solutions. John Wiley & Sons (1998)
6. Mundy, J., Thornthwaite, W., Kimball, R.: The Microsoft data warehouse toolkit. John Wiley & Sons (2006)
7. Song, I.Y., Khare, R., Dai, B.: SAMSTAR: a semi-automated lexical method for generating star schemas from an entity-relationship diagram. In: Proceedings DOLAP 2007. ACM, New York (2007)
8. Abello, A., Samos, J., Saltor, F.: Yam2: A multidimensional conceptual model extending UML. Information Systems 31(6), 541–567 (2006)
9. Kamble, A.: A Conceptual Model for Multidimensional Data. In: Asia-Pacific Conference on Conceptual Modelling - APCCM, pp. 29–38 (2008)
10. Moody, D., Kortink, M.: From ER models to dimensional models: bridging the gap between OLTP and OLAP design. Business Intelligence Journal 8(3) (2003)
11. Moody, D., Kortink, M.: From ER Models to Dimensional Models II: Advanced Design Issues. Business Intelligence Journal 8(4) (2003)
12. Payne, R.: MORM: a fact-oriented conceptual modeling approach to designing data warehouses and OLAP applications. MSc Thesis, University of British Columbia (2003), https://circle.ubc.ca/handle/2429/15349
13. Piprani, B.: Using ORM-Based Models as a Foundation for a Data Quality Firewall in an Advanced Generation Data Warehouse. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1148–1159. Springer, Heidelberg (2006)
14. Pierson, E.J., dela Cruz, N.: Using Object Role Modeling for Effective In-House Decision Support Systems. In: Chung, S., Herrero, P. (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 636–645. Springer, Heidelberg (2005)

15. Hansen, J., dela Cruz, N.: Evolution of a Dynamic Multidimensional Denormalization Meta Model Using Object Role Modeling. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1160–1169. Springer, Heidelberg (2006)
16. Halpin, T., Carver, A., Owen, K.: Reduction Transformations in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 699–708. Springer, Heidelberg (2007)
17. Halpin, T.: Temporal Modeling and ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 688–698. Springer, Heidelberg (2008)
18. de Brock, E.: Transactions in ORM. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) Enterprise, Business-Process and Information Systems Modeling. LNBIP, vol. 29, pp. 291–301. Springer, Heidelberg (2009)
19. Balsters, H., Carver, A., Halpin, T., Morgan, T.: Modeling Dynamic Rules in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1201–1210. Springer, Heidelberg (2006)
20. Balsters, H., Halpin, T.: Formal Semantics of Dynamic Rules in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 699–708. Springer, Heidelberg (2008)
21. Microsoft Corporation, `http://msdn.microsoft.com/en-us/library/ms141715.aspx`
22. Oracle Corporation, `http://download.oracle.com/docs/cd/E10926_01/doc/owb.101/b12151/apascd.htm`
23. Embarcadero Technologies, `http://docs.embarcadero.com/products/er_studio/erstudio8.0/erstudio802/ERStudioUserGuide.pdf`
24. van Griethuysen, J. (ed.): ISO Technical Report on Concepts and Terminology for the Conceptual Schema and the Information Base. ISO/TR 9007:1987, International Standards Organization, Geneva (1987)

Enriched Support for Ring Constraints

Terry Halpin[1] and Matthew Curland[2]

[1] LogicBlox, Australia and INTI International University, Malaysia
[2] LogicBlox, USA
{terry.halpin,matt.curland}@logicblox.com

Abstract. Fact-oriented modeling approaches such as Object-Role Modeling (ORM) have long supported several varieties of ring constraints, such as irreflexivity, asymmetry, intransitivity, and acyclicity, on pairs of compatible roles. The latest version of the Web Ontology Language (OWL 2) supports five kinds of ring constraint on binary predicates. Recently, three more ring constraint types (local reflexivity, strong intransitivity, and transitivity) were added to ORM. This paper discusses these new additions to ORM, as implemented in the Natural ORM Architect (NORMA) tool, and identifies important ways in which ORM and OWL differ in their support for ring constraints, while noting different mapping alternatives. We determine which combinations of elements from this expanded set of ring constraints are permitted, and provide verbalization patterns for the new additions. Graphical shapes for the new constraints and constraint combinations are introduced and motivated, and NORMA's new user interface for entry of ring constraints is illustrated.

1 Introduction

Data modeling approaches have long recognized the importance of integrity constraints to help ensure that data entered into the information system is consistent with the business domain being modeled. At the implementation level, such constraints are captured in code in a language relevant to that logical data model. For example, in a relational database the constraints are coded in SQL, in a semantic Web ontology the constraints might be coded in the *Web Ontology Language* (*OWL*) [20], and in a deductive database the constraints might be declared in datalog [1]. However, to ensure that the constraints enforced in the implementation code correspond to actual constraints in the universe of discourse, the constraints should first be specified in a conceptual schema, where they can be reliably validated with the business domain experts using concepts and language that are intelligible to them, and then transformed automatically into implementation code.

One class of constraints that is commonly found in business domains but less often captured in industrial data models deals with logical restrictions on how *pairs of compatible fact roles* may be populated. These constraints are called *ring constraints*, since in the most common case the role pairs form a *ring predicate*, where each role in the relationship is played by instances of the same type (see Figure 1(a)). Here a path from the type *A* through the predicate *R* and back to *A* intuitively forms a ring.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 309–318, 2011.
© Springer-Verlag Berlin Heidelberg 2011

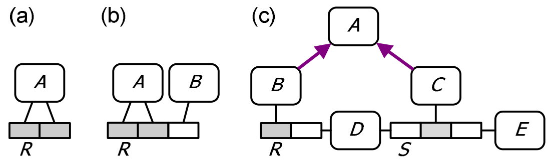


Fig. 1. Ring constraints may apply only to a pair of compatible roles

The schemas shown in Figure 1 use the notation of *Object-Role Modeling* (*ORM*) a conceptual approach to data modeling that structures the facts of interest in terms of *objects* (entities or values) that play *roles* (parts in relationships, which are logical *predicates*). Object types are graphically depicted as named, soft rectangles, and predicates as ordered sets of role boxes, along with a predicate reading, where each role is connected to the type whose instances may play that role. Subtyping is displayed using directed line segments. For example, in Figure 1(c), the object types *B* and *C* are subtypes of *A*. If types overlap (i.e. share a common instance), their roles are compatible. For explanation purposes, Figure 1 shades the roles that are compatible, and hence may be restricted by a ring constraint. By default, subtypes of a common supertype may overlap, thus the roles attached to *B* and *C* in Figure 1(c) are compatible.

Unlike attribute-based approaches such as Entity-Relationship (ER) modeling [2] and class diagramming within the Unified Modeling Language (UML) [17], ORM is *fact-oriented*, where all facts, constraints, and derivation rules may be verbalized naturally in sentences easily understood and validated by nontechnical business users using concrete examples. ORM's graphical notation for data modeling is far more expressive than that of industrial ER diagrams or UML class diagrams, and its attribute-free nature makes it more stable and adaptable to changing business requirements. Brief introductions to ORM may be found in [6, 9], a detailed introduction in [10], and thorough treatment in [13]. An overview of fact-oriented modeling approaches, including ORM and others such as natural Information Analysis Method (NIAM) [19], and the Predicator Set Model (PSM) [15], as well as history and research directions, may be found in [7]. In the 1980s, one of us formalized and refined an early version of NIAM, adding support for three kinds of ring constraint (irreflexivity, asymmetry and intransitivity) [4], and in later years evolved the approach into ORM, adding support for three more ring constraints (symmetry, acyclicity, and pure reflexivity).

The first version of OWL supported two kinds of ring constraint (symmetric and transitive) on binary predicates. The latest version of OWL (OWL 2) [20, 21, 23] added support for reflexive, irreflexive, and asymmetric ring constraints. In 2011, we extended ORM with support for three further ring constraints (strong intransitivity, local reflexivity, and transitivity). This paper discusses these new additions to ORM, as implemented in the Natural ORM Architect (NORMA) tool [3], and identifies important ways in which ORM and OWL differ in their support for ring constraints, while noting different mapping alternatives. We determine which combinations of elements from this expanded set of ring constraints are permitted, and provide verbalization patterns for the new additions. Graphical shapes for the new constraints and constraint combinations are introduced and motivated, and NORMA's new user interface for entry of ring constraints is illustrated.

The rest of this paper is structured as follows. Section 2 contrasts the different ways in which ORM and OWL address reflexivity constraints. Section 3 motivates the need for strong intransitivity, as supported in ORM, and discusses different ways to cater for transitivity. Section 4 identifies the ways in which the ring constraint primitives may be combined, and discusses NORMA's new user interface for coping with the large number of possible combinations. Section 5 summarizes the main results, outlines future research directions, and lists references.

2 Reflexive and Irreflexive Constraints

In mathematics, a relation R on a set A is said to be *reflexive* over A if and only if xRx for each element x in A. For example, the relation $\leq$ on the set of real numbers is reflexive and the subsethood relation $\subseteq$ is reflexive, since every set is a subset of itself. A relation R is *globally reflexive* if and only if every individual thing in the domain of discourse bears the relation R to itself (i.e., R is globally reflexive if and only if $\forall x\ xRx$). For example, in a world that includes numbers and sets, the identity relation = is globally reflexive, but $\leq$ is not globally reflexive because $\leq$ does not apply to sets.

OWL allows object property expressions to be declared globally reflexive by characterizing them as reflexive properties [21, p. 10]. For example, if we restrict the domain of individuals (owl:Thing) to persons, and we agree that each person knows himself/herself, then the "knows" predicate may be declared to be reflexive in Turtle syntax thus: :knows rdf:type owl:ReflexiveProperty. In practical business domains, global reflexivity rarely applies to a predicate because the domain typically includes many types of individuals (e.g. if the world includes both persons and projects, then "knows" is not globally reflexive because "knows" does not apply to projects). To partly address this problem, OWL includes an ObjectHasSelf restriction to declare a subset of a predicate's domain where each individual in that subset bears the relation to itself. For example, we may define SelfLiker as a subclass of Person, where each person in SelfLiker does like himself/herself. In Manchester syntax, if likes is an object property, this restriction may be declared thus: Class: SelfLiker EquivalentTo: likes Self.

In the interests of wider applicability, ORM defines reflexivity in the following local sense. A ring predicate R is *locally reflexive* if and only if $\forall x \forall y(xRy \rightarrow xRx)$. So, if an individual bears the relation to something, then it must bear the relation to itself. Hence in ORM you may declare the predicate "knows" to be locally reflexive even when the universe of discourse includes objects like projects where the predicate does not apply at all. ORM can also support the mathematical notion of reflexivity over a set by adding a mandatory role constraint. For example, if knows is locally reflexive and we constrain each person to know somebody then knows is reflexive over people. ORM also supports purely reflexive constraints. A ring predicate R is *purely reflexive* if and only if $\forall x \forall y(xRy \rightarrow x = y)$. Unlike OWL, ORM allows ring constraints to be applied to compatible role pairs in general, not just binary predicates. Finally, ORM supports both alethic and deontic modalities for all its constraints [7]. It should now be clear that ORM's support for reflexivity is much deeper than that of OWL.

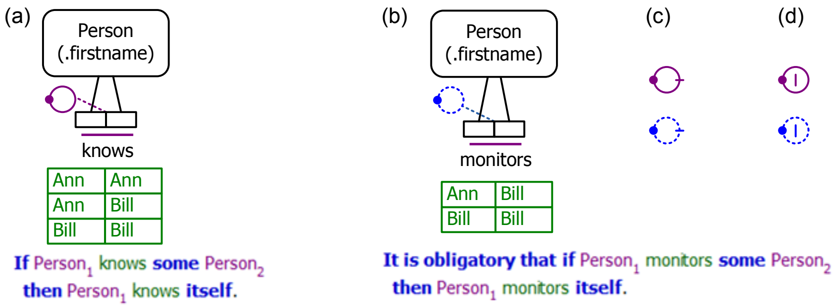


Fig. 2. Locally reflexive, irreflexive, and purely reflexive ring constraints in ORM

A binary relation *R* is *irreflexive* if and only if $\forall x \sim xRx$. For example, isParentOf is irreflexive since nobody is his/her own parent. OWL supports irreflexive constraints on binary relations, and ORM supports them on compatible role-pairs.

Unlike OWL (and ER and UML), ORM depicts ring constraints graphically. Moreover, the NORMA tool automatically *verbalizes* constraints so they can be easily validated with nontechnical users [11]. Local reflexivity is depicted by a circle with a dot, connected to the constrained roles, as in Figure 2(a). The dot denotes an object, and the circle denotes a relationship from the object to itself. For the deontic version, the color is blue instead of violet and the circle is dashed, as in Figure 2(b), where the sample population is possible even though it violates the deontic constraint. The irreflexivity shape adds a stroke through the circle, denoting negation of the relationship (Figure 2(c)). The purely reflexive constraint shape (d) adds a stroke in the inner area, suggesting negation of all non-reflexive relationships. The bottom text in Figures 2 (a) and (b) shows NORMA's verbalization of the alethic and deontic ring constraints.

3 Transitive, Intransitive, and Strongly Intransitive Constraints

A ring predicate *R* is *transitive* if and only if $\forall x,y,z[(xRy \ \& \ yRz) \rightarrow xRz]$. For example, the isAncestorOf is transitive because if one person is an ancestor of another, and the second person is an ancestor of a third person, then the first person is an ancestor of the third person. Figure 3(a) shows an ORM diagram of the ancestorhood fact type, together with a sample population. ORM depicts a transitivity constraint by a triangle with dots denoting objects at each node, as in Figure 3(b). The lines denote relationships (if the lefthand object relates to the top object, and the top object relates to the righthand object, then the lefthand object relates to the righthand object). The transitivity constraint in Figure 3(a) verbalizes thus: **If** Person_1 is an ancestor of Person_2 **and** Person_2 is an ancestor of Person_3 **then** Person_1 is an ancestor of Person_3.

Ignoring reincarnation, the ancestorhood predicate is *acyclic* (an object can't cycle back to itself via one or more ancestorhood facts). Ancestorhood is also asymmetric, but this is implied by acyclicity. ORM depicts the acyclic constraint graphically by a ring with three dots and a stroke, as in Figure 3(c). These two constraint shapes may be orthogonally combined by overlaying into a single icon as in Figure 3(a).

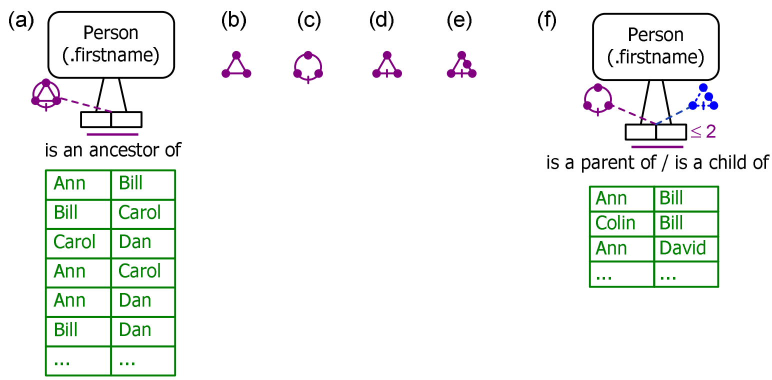


Fig. 3. Transitive, acyclic, intransitive and strongly intransitive constraints in ORM

Although OWL does not support acyclicity constraints, it does allow ring predicates to be declared to be transitive and asymmetric. For example, in Manchester syntax, the isAncestorOf object property may be declared to have the characteristics "Transitive, Asymmetric".

A ring predicate R is *intransitive* if and only if $\forall x,y,z[(xRy \ \& \ yRz) \rightarrow \sim xRz]$. The ORM intransitive constraint shape adds a stroke (indicating negation) to the bottom line of the transitive constraint shape, as in Figure 3(d). In practice, most intransitive relationships are strongly intransitive. In the early 1990s, support for strong intransitivity was proposed by Ritson and Halpin, under the name "intransitive jump" [16, p. 3-19], but we only recently implemented this in NORMA. Strong intransitivity blocks jumps over one or more nodes, so includes intransitivity as a special case (jumping over one node only). Like acyclicity, strong intransitivity involves recursion.

A ring predicate R is *strongly intransitive* if and only if $\forall x,y,z[(xRy \ \& \ yPz) \rightarrow \sim xRz]$, where the predicate P is recursively defined to give the transitive closure of R, i.e. $\forall x,y[xPy \leftarrow (xRy \vee \exists z(xRz \ \& \ zPy))]$. While strong intransitivity may be enforced in SQL using recursive union, datalog enables a more elegant implementation. For example, in Datalog^{LB}, where "<-", ";", "," and "!" denote "**if**", "**or**", "**and**" and "**not**" respectively, a strong intransitivity constraint on parenthood may be enforced by deriving the isAncestorOf predicate as the transitive closure of the isParentOf predicate, thus

```
isAncestorOf(x,y) <- isParentOf(x,y) ; isParentOf(x,z), isAncestorOf(z,y).
```

and then applying the following further constraint

```
isParentOf(x,y), isAncestorOf(y,z) -> !isParentOf(x,z).
```

The NORMA tool has recently been extended to map ORM models to Datalog^{LB}, not just SQL. Datalog^{LB} is a vastly extended version of datalog [1] that employs fact-oriented data structures with performance benefits similar to those of column stores when processing complex rules over large data sets. Datalog^{LB} extends basic datalog with stratified negation, types, functions (including aggregate functions), transactions, modules, constraints, default values, ordered predicates, metalevel support, and other features. An overview of mapping ORM to Datalog^{LB} may be found in [12]. While NORMA supports mapping of many ORM constraints to either SQL or Datalog^{LB}, mapping of the ring constraints discussed in this paper is yet to be implemented.

ORM's graphical shape for strong intransitivity adds an extra node to the basic intransitivity shape, as in Figure 3(e), suggesting the path may be longer. In Figure 3(f), parenthood is declared to be alethically acyclic (no reincarnation) and deontically, strongly intransitive (incest is possible but forbidden). The deontic, strong intransitivity constraint verbalizes thus: **It is obligatory that if** $Person_1$ is a parent of **some** $Person_2$ **then it is not true that** $Person_1$ **is indirectly related to** $Person_2$ **by repeatedly applying this fact type**.

In the ORM model shown in Figure 3(a), ancestorhood facts are simply asserted. If instead, ancestorhood facts are always derivable from parenthood facts, then ORM can express this formally by deriving the ancestorhood fact type from this derivation rule: $Person_1$ is an ancestor of $Person_2$ **if and only if** $Person_1$ is a parent of $Person_2$ **or** $Person_1$ is a parent of **some** $Person_3$ **who** is an ancestor of $Person_2$. This FORML [14] rule is equivalent to the earlier datalog rule for ancesterhood, except the conditional is strengthened to a biconditional. Transitivity of ancestorhood is now implied by the derivation rule.

One reason for deriving transitive predicates rather than asserting them is that their population (or transitive closure) can quickly become very large. For example, asserting the top three facts in the sample data for Figure 3(a) requires the bottom three facts shown to be included. As the number of asserted facts increases, the number of facts that are transitively implied increases dramatically.

In OWL, declaring the isAncestorOf predicate to be transitive does not require that all its instances must be asserted. For example, if you declare the predicate to be transitive, and assert that Ann is an ancestor of Bill and that Bill is an ancestor of Chris, then the OWL engine can infer that Ann is an ancestor of Chris. You can also arrange things this way in ORM by declaring the ancestorhood fact type to be *semiderived*, and supplying the following derivation rule: $Person_1$ is an ancestor of $Person_2$ **if** $Person_1$ is a parent of $Person_2$ **or** $Person_1$ is a parent of **some** $Person_3$ **who** is an ancestor of $Person_2$. This weakens the earlier biconditional rule in FORML to a conditional (like OWL, ORM adopts the open world assumption by default). However, if all relevant parenthood facts are available to fully derive ancestry, then the earlier derivation rule discussed is clearly preferable.

4 Combining Ring Constraints

The recent addition of reflexive, strongly intransitive, and transitive ring constraints to ORM significantly increased the number of legal combinations of ring constraints that may be expressed. Candidate legal combinations were first formally determined by logic, to ensure that such combinations are *strongly satisfiable*. Strong satisfiability goes beyond simple logical consistency by requiring that each predicate must be populatable [4]. The list of strongly satisfiable combinations was then reduced to a more manageable number, by removing patterns that included a constraint derivable from other constraints in the combination. We then redesigned NORMA's user interface for entering ring constraints to facilitate the task of choosing ring constraint combinations.

(a) *Sym & Trans implies LRefl*

1.	$\forall xy(xRy \rightarrow yRx)$	Sym
2.	$\forall xyz[(xRy \,\&\, yRz) \rightarrow xRz]$	Trans
3.	$\sim\forall xy(xRy \rightarrow xRx)$	~Lrefl
4.	$\exists$xy~(xRy → xRx)	3 QN
5.	$\sim(aRb \rightarrow aRa)$	4 EI
6.	aRb	5 PC
7.	$\sim aRa$	5 PC
8.	$aRb \rightarrow bRa$	1 UI
9.	bRa	8, 6 AA
10.	$(aRb \,\&\, bRa) \rightarrow aRa$	2 UI
11.	$aRb \,\&\, bRa$	6, 9 Conj
12.	aRa	10, 11 AA
	×	12, 7

(b) *Lrefl & Irrefl is not StrSat*

1.	$\forall xy(xRy \rightarrow xRx)$	Lrefl
2.	$\forall x\sim xRx$	Irrefl
3.	$\exists xy\; xRy$	StrSat
4.	aRb	3 EI
5.	$aRb \rightarrow aRa$	1 UI
6.	aRa	5,4 AA
7.	$\sim aRa$	2 UI
	×	6, 7

Fig. 4. Deduction trees to determine the legality of two ring constraint combinations

Figure 4 shows deduction trees to determine the legality of two of the many ring constraint combinations investigated. Deduction trees augment sematic tableaux with natural deduction, and provide a convenient formal mechanism for establishing logical results in ORM [4]. Justification columns on the right provide the justification for each step (e.g. QN = Quantifier Negation, EI = Existential Instantiation, UI = Universal Instantiation, AA = Affirming the Antecedent). Figure 4(a) shows that a predicate that is symmetric and transitive must also be locally reflexive, hence there is no need to explicitly add local reflexivity if the former two constraints apply. Figure 4(b) shows that the combination of local reflexivity and irreflexivity is not strongly satisfiable—note that step 3 is needed, because the two constraints are compatible in a universe where the predicate *R* is never populated.

The introduction of strongly intransitive, transitive, and reflexive ring types increased the number of choices of single ring types from 7 to 10, paired ring types from 4 to 15, and added a valid triple combination (reflexive, transitive, and antisymmetric). The increase in the total number of choices made the old user interface approach of choosing from a fixed list of valid ring types extremely cumbersome. Apart from the unwieldy length of the list, the position of combinations in the list is unpredictable because the combinations are arbitrarily ordered. For example, the pair combination *symmetric* and *irreflexive* could be listed with equal validity beside either of its component ring types, which are alphabetically distant in a sorted dropdown list.

Any design for an alternative to the inclusive list must allow selection of any of the single ring types, so ten items is the minimum available list size. The user-interface must allow combinations, so a multi-select or checkbox approach to allow the user to select multiple ring types is also a requirement. However, the ability to freely choose combinations needs to be restricted given that fewer than a third of the pair combinations and only a single triple are valid.

We disallow ring patterns if the overlap is not strongly satisfiable (e.g. *symmetric* and *antisymmetric*) or if one ring type implies the other (e.g. *acyclic* implies *asymmetric* which in turn implies *antisymmetric* and *irreflexive*). As these two reasons for disallowing ring type pairs have opposite motivations, we want our user interface (UI) to reflect this difference. By displaying implications in the UI we also help the user to remember the assorted ring type implication patterns.

The NORMA ring type chooser does not disable any elements, thus allowing the user to move between different single and combination selections. We use two types of feedback: the checkbox state (checked, not checked, and gray checked), and font strength (bold and normal). A checked item is part of the current selection, and a gray checked item is implied. A bold item can be checked without removing other checked items, while checking a non-bold item removes at least one other check. For a state with at least one selected item, this gives us four possible item states: *bold+checked* indicates a currently selected ring type, *bold+unchecked* indicates a ring type that is compatible with the current selection, *gray checked+normal* indicates an implied item (checking this item will uncheck the implying type), and *unchecked+normal* indicates an incompatible item. Examples of each of these item states are shown in Figure 5.

The final decision for the ring type chooser design concerned the display order. The chosen grouping places positive ring types (reflexive, symmetric, and transitive) after the negative ring types. The subgroups are then ordered by the somewhat subjective category of strength, with stronger constraints appearing last. The result is that negative ring types imply types above them in the list. The exception to most of these patterns is *purely reflexive*, which is a negative ring type that implies positive types. The result is a dynamic user-interface with immediate feedback on ring type compatibility and implication that provides strong guidance and facilitates user learning by showing implication patterns and associating the single ring types with the corresponding glyphs.

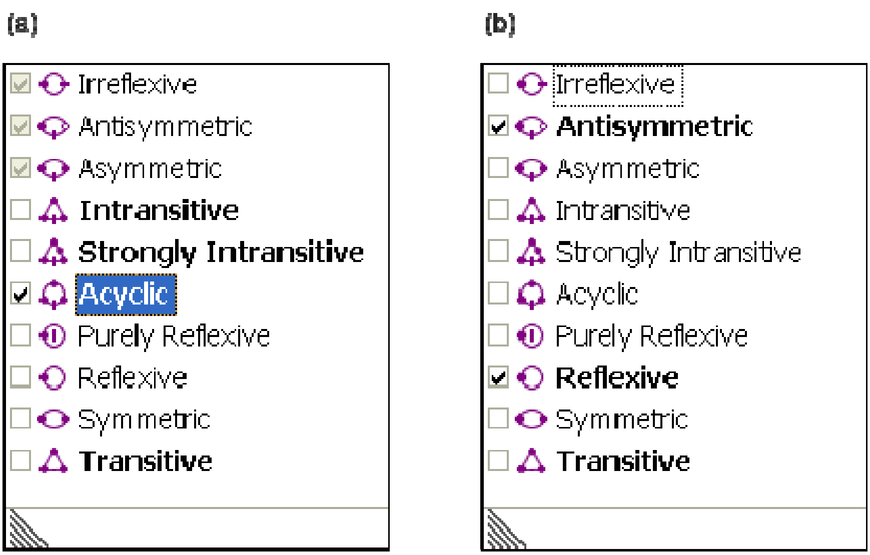


Fig. 5. Ring type chooser indicating an acyclic ring type (a) and reflexive+antisymmetric combination ring type (b)

5 Conclusion

This paper discussed our recent work to extend ORM's support for ring constraints by adding locally reflexive, strongly intransitive, and transitive ring constraints. The differences between ORM and OWL in this regard were examined, and ORM's graphical notation for, and NORMA's automated verbalization of, the new ring constraints was illustrated. Legal combinations of the expanded set of ring constraints were identified, and some sample proofs provided for these results. Finally, NORMA's new user interface for entry of ring constraints was discussed.

While NORMA now supports rich graphical support and automated verbalization for ring constraints, full support for transformation of these conceptual constraints into implementation code in target systems such as relational and deductive databases has yet to be completed.

As discussed in [13], many ring constraints (e.g. irreflexive, asymmetric, antisymmetric, intransitive, strongly intransitive, and acyclic) are "negative' in the sense that they are designed to prevent entry of illegal facts. In contrast, reflexive, symmetric and transitive characteristics are "positive" in the sense that they could be used to infer further facts (as in OWL). In implementing the mapping of these "positive" characteristics in NORMA, we plan to provide flexible options for users. For example, if the user chooses to treat local reflexivity as a constraint, we plan to provide an option to automatically insert the reflexive instances, thus reducing the data entry effort for the user (similarly for symmetry and transitivity). We also plan to facilitate the implementation of such positive characteristics (especially transitivity) via semiderivation rather than simply constraining fully asserted fact types.

Basic work on mapping from ORM to OWL has been undertaken by several researchers. Various theoretical results in this regard are published in the literature, and an early prototype using NORMA to do a basic mapping has been completed. However, much more work needs to be done to provide an industrial strength mapping from ORM to OWL. Part of the motivation for adding support for reflexivity and transitivity constraints to ORM, including the planned mapping choices sketched in the previous paragraph, is to facilitate such a mapping.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Chen, P.P.: The entity-relationship model—towards a unified view of data. ACM Transactions on Database Systems 1(1), 9–36 (1976), `http://csc.lsu.edu/news/erd.pdf`
3. Curland, M., Halpin, T.: The NORMA Software Tool for ORM 2. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 190–204. Springer, Heidelberg (2011)
4. Halpin, T.: A Logical Analysis of Information Systems: static aspects of the data-oriented perspective. Doctoral dissertation, University of Queensland (1989), `http://www.orm.net/Halpin_PhDthesis.pdf`
5. Halpin, T.: ORM 2. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 676–687. Springer, Heidelberg (2005)

6. Halpin, T.: ORM/NIAM Object-Role Modeling. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Information Systems Architectures, 2nd edn., pp. 81–103. Springer, Heidelberg (2006)
7. Halpin, T.: Modality of Business Rules. In: Siau, K. (ed.) Research Issues in Systems Analysis and Design, Databases and Software Development, pp. 206–226. IGI Publishing, Hershey (2007)
8. Halpin, T.: Fact-Oriented Modeling: Past, Present and Future. In: Krogstie, J., Opdahl, A., Brinkkemper, S. (eds.) Conceptual Modelling in Information Systems Engineering, pp. 19–38. Springer, Berlin (2007)
9. Halpin, T.: Object-Role Modeling. In: Liu, L., Tamer Ozsu, M. (eds.) Encyclopedia of Database Systems. Springer, Berlin (2009)
10. Halpin, T.: Object-Role Modeling: Principles and Benefits. International Journal of Information Systems Modeling and Design 1(1), 32–54 (2010)
11. Halpin, T., Curland, M.: Automated Verbalization for ORM 2. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1181–1190. Springer, Heidelberg (2006)
12. Halpin, T., Curland, M., Stirewalt, K., Viswanath, N., McGill, M., Beck, S.: Mapping ORM to Datalog: An Overview. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 504–513. Springer, Heidelberg (2010)
13. Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann, San Francisco (2008)
14. Halpin, T., Wijbenga, J.: FORML 2. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 247–260. Springer, Heidelberg (2010)
15. ter Hofstede, A., Proper, H., van der Weide, T.: Formal definition of a conceptual language for the description and manipulation of information models. Information Systems 18(7), 489–523 (1993)
16. Ritson, P.: Use of Conceptual Schemas for a Relational Implementation. Doctoral dissertation, University of Queensland (1994)
17. Object Management Group: UML 2.0 Superstructure Specification (2003), `http://www.omg.org/uml`
18. Object Management Group: UML OCL 2.0 Specification (2005), `http://www.omg.org/docs/ptc/05-06-06.pdf`
19. Wintraecken, J.: The NIAM Information Analysis Method: Theory and Practice. Kluwer, Deventer (1990)
20. W3C: OWL 2 Web Ontology Language: Primer (2009), `http://www.w3.org/TR/owl2-primer/`
21. W3C: OWL 2 Web Ontology Language: Direct Semantics (2009), `http://www.w3.org/TR/owl2-direct-semantics/`
22. W3C: OWL 2 Web Ontology Language Manchester Syntax (2009), `http://www.w3.org/TR/owl2-manchester-syntax/`
23. W3C: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (2009), `http://www.w3.org/TR/owl2-syntax/`

Value Grouping Transformations in ORM

Ron McFadyen

University of Winnipeg, Winnipeg, Manitoba, Canada
r.mcfadyen@uwinnipeg.ca

Abstract. This paper proposes an extension to the Value Concatenation / Value Separation schema transformation defined for Object-Role Modeling. The basic transformation can be applied to convert a many-to-many fact type to a functional fact type. When applied, the transformation generates a new multi-valued object type where values are derived from an existing single-valued object type. Motivated by techniques applied in dimensional modeling applications, Value Grouping considers the various options available in the grouping process and is seen as an extension to the basic Value Concatenation / Value Separation schema transformation.

Keywords: object-role model, equivalence transformations, value concatenation, value separation, dimensional modeling, many-to-many relationship.

1 Introduction

Dimensional modeling (DM) [1-2] is a database design technique used extensively in the design and construction of data warehouses [3]. At its simplest, a relational database implementing a dimensional model has one table designated as the *fact* table which participates in many-to-one relationships with other tables known as *dimension* tables. The term *fact* is used differently in DM than how *fact* is used in an object-role modeling (ORM) [4] context. Where necessary, we will differentiate in usage through the terms *fact table* (DM) and *fact type* (ORM).

To account for different requirements there are several standard techniques data warehouse designers may utilize in their dimensional designs such as outriggers, mini-dimensions, multi-valued attributes, helper tables, etc. Our focus in this paper is on techniques a designer may use to accommodate a many-to-many relationship between a fact table and a dimension table.

A complete treatment of ORM is found in [4]; several publications such as [5-8] relate to ORM-based warehousing. When the database analyst is creating a database design from user requirements several alternative schemas may need to be considered before a final choice is made. ORM is useful in this regard because there is some theoretical basis for recognizing certain schemas are equivalent to one another. [4, Ch. 14] discusses several results relating to ORM schema transformations and equivalence. In this paper our focus is on schema transformations concerning many-to-many fact types.

The rest of this paper is organized as follows. We begin in Section 2 with a review of dimensional modeling and some standard techniques for handling many-to-many

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 319–328, 2011.
© Springer-Verlag Berlin Heidelberg 2011

relationships between a fact table and a dimension table. Section 3 discusses the ORM Value Concatenation / Value Separation transformation followed by its generalization in Section 4. Lastly, we summarize our results and consider future directions.

2 Dimensional Modeling

DM is a constrained form of entity-relationship modeling which results in relational databases where a central fact table participates in several many-to-one relationships with dimension tables. The fact table contains performance measurements of events in a business process (e.g. acquisition value, appraisal value, width, height, etc. for letter facts). The dimension tables are characterized as having attributes that are descriptive (a keyword category, the date when a letter was appraised, etc.) and typically are not normalized. When combined with a fact table, dimension tables provide the descriptive information for events. In general dimensional designs are considered to be much simpler for business analysts to work with compared to normalized designs that exist in source systems. Figure 1 illustrates a dimensional model for a letter archive. Note how some dimensions play many roles appearing in many relationships.

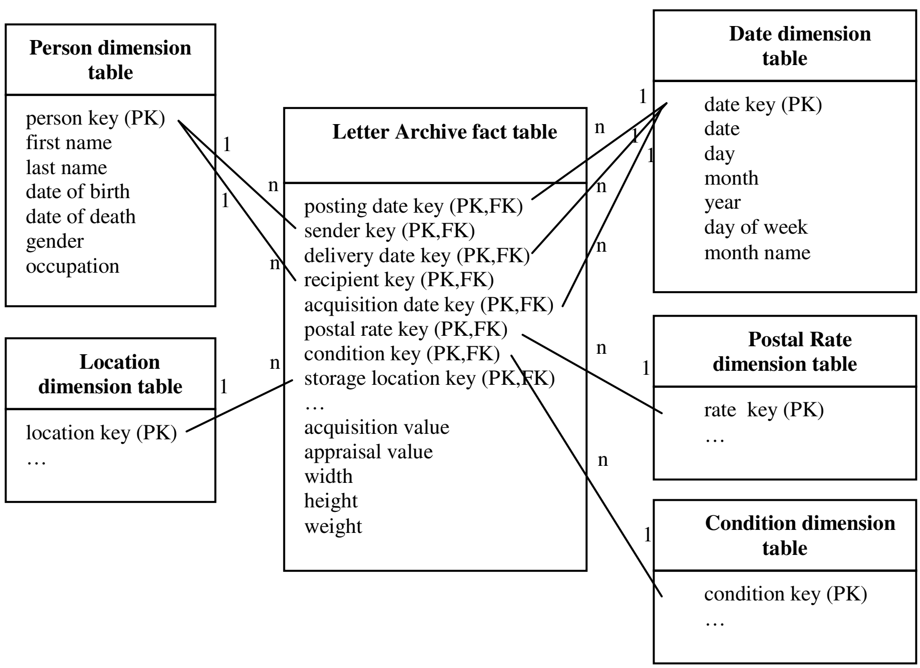


Fig. 1. A basic dimensional design for a letter archive warehouse

In the basic dimensional database a fact table is related to dimension tables via mandatory many-to-one relationships (foreign keys in the fact table cannot be null). Dimension tables have a primary key (PK) that is a surrogate key; DM purposely avoids other approaches that involve composite keys where a natural key is used in conjunction with some date and/or time related attribute. The PK for a fact table typically comprises several (possibly all) foreign keys relating to dimension tables.

[1-2], [9-10] introduce problems in various contexts and provide practical solutions as guidance for the warehouse designer. Many examples are given of situations where a dimension table has a natural many-to-many relationship with the fact table. The standard approach for handling a many-to-many relationship [11] in a relational database is to create an intersection table where the PK of the intersection table comprises the two PKs of the related tables. Because the primary key of the fact table can comprise all the foreign keys, this could result in a rather large composite key for the intersection table. This solution is not recommended in practice. Below, we present two examples of preferred approaches.

Example 1. In a health care example [9], a health care billing fact table and a diagnosis dimension table are related via a many-to-many relationship (a patient can have several diagnoses). Four alternative models are considered - the preferred solution is given in Figure 2.

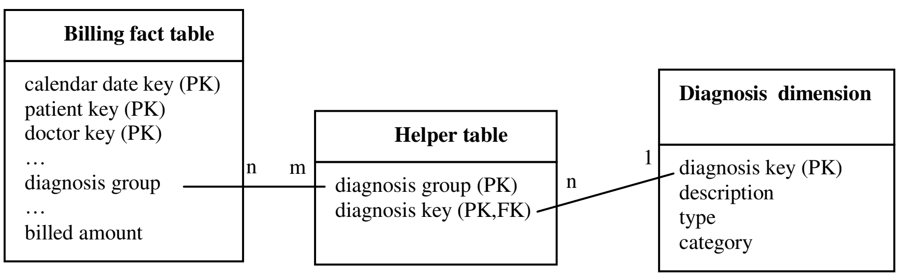


Fig. 2. Preferred solution for m:n relationship for billing fact table and a diagnosis dimension table

This is not a standard relational solution for resolving a many-to-many relationship; the *helper* table has a primary key comprising a group identifier and a diagnosis key. The *helper* table organizes diagnoses into groups, and the group identifier attribute, *diagnosis group*, is used to specify which diagnoses form a complex diagnosis for a patient. The relationship between *Billing fact* table and *Helper* table is many-to-many. An interesting point is that the *diagnosis group* attribute in the *Billing fact* table is not a foreign key – the attribute is just one part of the *Helper* table's primary key. A group of rows in the *Helper* table are related to one row in the *Billing fact* table. Figure 2 represents a practical solution with benefits related to incorporating additional information such as weighting factors for diagnoses within a group. The interested reader will find more discussion in [9].

Example 2. [10] describes an example with a massive letter archive where there are many characteristics describing letters such as sender, recipient, date sent, etc. – all of which fit nicely into many-to-one relationships. Additionally there are keywords that can be applied to letters and this leads to a many-to-many relationship. Different solutions are considered - Figure 3 presents the preferred solution and one alternate solution.

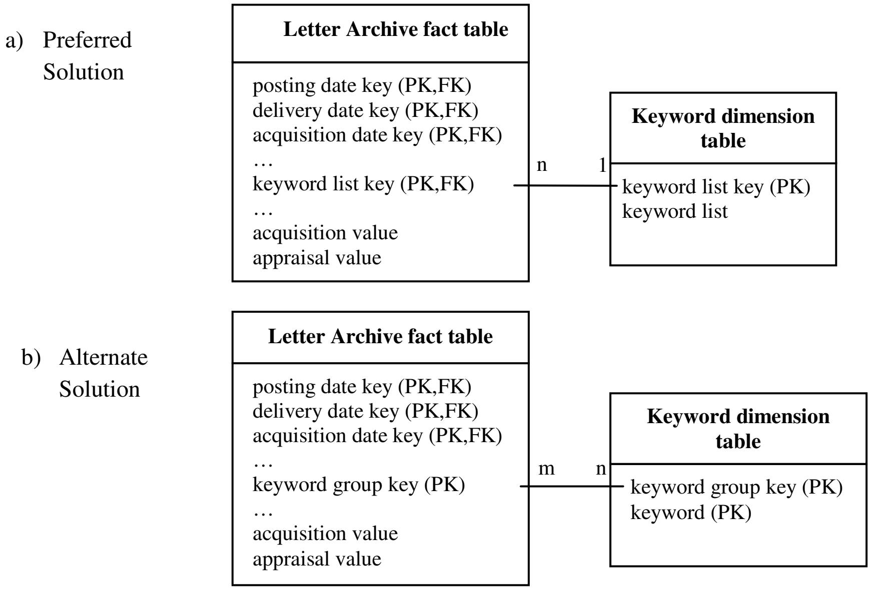


Fig. 3. Solutions for many-to-many relationship involving letters and keywords

Neither Figure 3.a) nor 3.b) is a standard relational solution for resolving many-to-many relationships. Some points of interest are: (1) regarding Figure 3.a) the Keyword dimension table has a multi-valued attribute *keyword list*, and (2) regarding Figure 3.b) the Keyword dimension table has a composite PK, the attribute *keyword* is single-valued, and the attribute *keyword group key* in the Letter Archive fact table is not a foreign key.

Both solutions in Figure 3 organize keywords into groups. In the preferred solution (Figure 3.a) the attribute *keyword list* is multi-valued and assumes values such as "Civil War", "Train Wreck", "Civil War/Train Wreck" where "/" is the chosen delimiter for values. In contrast, the *keyword* attribute in Figure 3.b) is single-valued with such values as "Civil War" and "Train Wreck", but not "Civil War/Train Wreck".

Figure 3.b) is structurally similar to Figure 2 with respect to the way the grouping is accomplished. In Figure 3.b) there are no attributes describing keywords, but it would be a simple matter to extend the example if we were to include attributes such as keyword category, language, source (author/automated), etc. Similarly Figure 3.a) could be extended with a Keyword table to provide additional attributes for keywords.

[10] contrasts the two solutions for Figure 3 according to the ease of formulating SQL queries. For example, suppose an end-user wants to find documents relating to keywords Civil War and Train Wreck. In one case suppose the user wants to find all documents relating to both Civil War <u>and</u> Train Wreck, and in the other case to find all documents relating to either Civil War <u>or</u> Train Wreck. Conceptually these two queries, posed in English, are very similar in structure. But when one considers the SQL statements (see Table 1 for sample SQL) to produce results for these two cases the SQL statements are similar to one another for Figure 3.a), but very different for

Figure 3.b). When one examines the SQL based on Figure 3.a) one can easily see the symmetry in the code; the two SELECT statements differ only in the operators OR and AND. But when one examines the SQL based on Figure 3.b) one easily sees the simple 'OR' query and a complicated 'AND' query. There are several ways to code the 'AND' query – the code given is based on [10]. The important aspect of the query is that it is non-trivial, perhaps beyond the skill level of an end-user. One objective of DM is to produce database designs that are easy for a user to work with, and so based on a comparison of SQL statements Figure 3.a) is preferred.

Table 1. Sample SQL for AND/OR queries pertaining to Figure 3

	Figure 3.a)	Figure 3.b)

OR	SELECT L.*
	FROM LetterArchive L
	INNER JOIN Keyword K
	ON (L.keywordListKey=
	    K.keywordListKey)
	WHERE
	keywordList like
	'%Civil War%'
	OR
	keywordList like
	'%Train Wreck%'
		SELECT L.*
		FROM LetterArchive L INNER JOIN
		Keyword K1
		ON (L.keywordGroupKey=
		    K1.keywordGroupKey)
		WHERE
		keyword = 'Civil War'
		OR
		keyword = 'Train Wreck'

AND	SELECT L.*
	FROM LetterArchive L
	INNER JOIN Keyword K
	ON (L.keywordListKey=
	    K.keywordListKey)
	WHERE
	keywordList like
	'%Civil War%'
	AND
	keywordList like
	'%Train Wreck%'
		SELECT L.*
		FROM LetterArchive L INNER JOIN
		Keyword K1
		ON (L.keywordGroupKey=
		    K1.keywordGroupKey)
		WHERE
		 (SELECT COUNT(K2.keywordGroupKey)
		  FROM Keyword K2
		  WHERE K1.keywordGroupKey=
		        K2.keywordGroupKey
		  AND
		  K2.keyword in
		     ('Civil War', 'Train Wreck')
		  GROUP BY K2.keywordGroupKey) **= 2**

3 ORM Schema Transformations

Given a set of user requirements there may be several schemas that fulfill the requirements; each may have its own advantages and disadvantages that the modeler must consider. [4, Ch. 14] discusses equivalent ORM schemas and schema transformations. Of interest here is a transformation referred to as Value Concatenation / Value Separation (VC/VS) that can be used to transform a many-to-many binary fact type to a functional fact type. The example given in [4] relates to the training of physicians. If we consider that a physician can be trained in medicine, acupuncture, or both, then we know there are 3 training possibilities for a physician. Figure 4 illustrates two equivalent schemas. Consider a physician trained in both medicine and acupuncture; in Figure 4.a) there would be two instances of the *is trained in* fact type for that physician, but in Figure 4.b) there would be only one instance of the *is trained in* fact type for that same physician.

a) With a single-valued object type

b) With a multi-valued object type

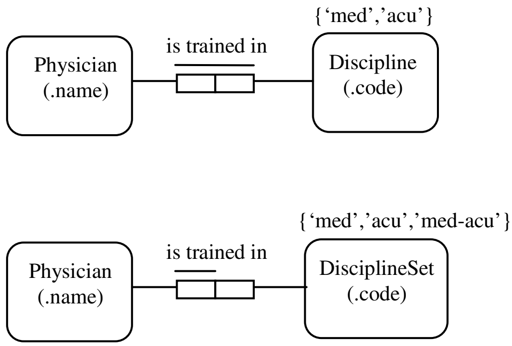


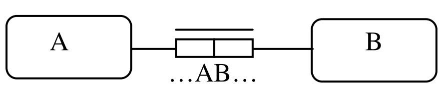
Fig. 4. Equivalent schemas via Value Concatenation / Value Separation

In general this transformation is considered of lesser importance. This transformation type may be considered less useful than others for a number of reasons: the transformation can become unwieldy due to the explosion of values if we need to consider all possible combinations of values; multi-valued object types require extra effort when coding for retrieval, updates, and deletions. In the example above we have chosen to use "-" as the delimiter for values in the multi-valued object type. If the cardinality of DisciplineSet were large then we may prefer to order (e.g. alphabetically) the values making up an instance.

The essential aspect of the transformation from Figure 4.a) to Figure 4.b) is the grouping of disciplines. If a physician is trained in more than one discipline then the instance is determined by concatenating the pertinent values with a defined delimiter such as "-".

4 Value Grouping

In section 2 we reviewed DM techniques where a model is transformed through grouping. In section 3 we reviewed the ORM VC/VS transformation where grouping



a1	b1
a1	b2
a2	b1
a2	b2
a3	b3
a4	b4
a5	b2
a5	b3

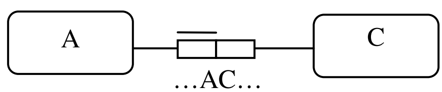
Fig. 5. m:n fact type

is used to generate an equivalent schema. From this point on we discuss schemas using ORM terminology. In this section we discuss two Value Grouping (VG) transformations that represent VC/VS and DM techniques; lastly, we introduce a potential schema annotation for fact types that are not derived and not stored.

We see two fundamental choices to find a way to group and organize instances based on either multi-valued instances or single-valued instances. Figure 5 presents a generic m:n fact type where object types A and B play roles, and where any value constraints concerning A and B are based on single-valued instances.

Choice 1. Figure 6 illustrates two models based on multi-valued instances that are equivalent to Figure 5. Figure 6.a) is just the VS/VC transformation from Figure 4, section 3. Figure 6.b) is a simple transformation applied to Figure 6.a) where a reference scheme identifies instances. Note that for Figure 6.b), the NORMA [12] plugin for Microsoft Visual Studio 2010 generates a database with the same structure as Kimball's preferred solution for a keyword dimension shown in Figure 3.a) of section 2.

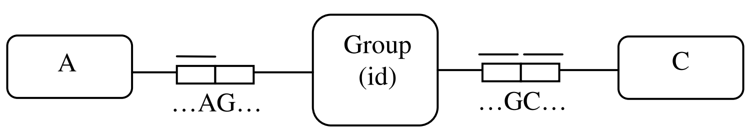
a)



a1	b1, b2
a2	b1, b2
a3	b3
a4	b4
a5	b2, b3

Value Constraint:
a,b appears in fact type AB iff a,c appears in fact type AC where c is a multi-value in C comprising single-values c_1, $c_{2,} \ldots$ c_n where exactly one $c_i = b$.

b)



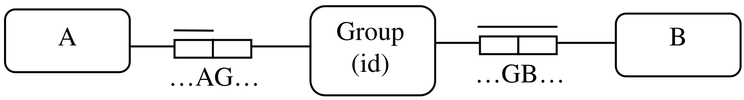
a1	g1
a2	g1
a3	g2
a4	g3
a5	g4

g1	b1, b2
g2	b3
g3	b4
g4	b2, b3

Value Constraint:
a,b appears in fact type AB iff a,g appears in fact type AG and g,c appears in fact type GC where c is a multi-value in C comprising single-values c_1, $c_{2,} \ldots$ c_n where exactly one $c_i = b$.

Fig. 6. Equivalent models based on multi-valued instances

Choice 2. Consider the schema in Figure 7 that is also equivalent to the schema in Figure 5. This schema utilizes the existing single-valued object type. This schema represents the DM techniques illustrated in Figure 2 and Figure 3.b). Note that for Figure 7, the NORMA plugin for Microsoft Visual Studio 2010 generates a database with the same structure as Figure 3.b).



a1	g1
a2	g1
a3	g2
a4	g3
a5	g4

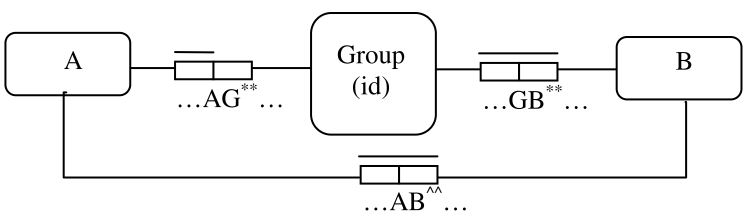
g1	b1
g1	b2
g2	b3
g3	b4
g4	b2
g4	b3

Value Constraint:
a,b appears in the fact type AB iff a,g appears in the fact type AG and g,b appears in fact type GB

Fig. 7. Equivalent schema based on single-valued instances

Figures 6 and 7 illustrate the two fundamental forms for the VG transformation: (1) use a multi-valued object type with or without a reference scheme, (2) use a single-valued object type and introduce a reference scheme. These two transformations cover the ORM VS/VC transformation and the DM techniques.

Lastly, in Figure 8 we introduce a (potential) new ORM notation that allows us to present an original schema (Figure 5) and its transformation (Figure 7) together; we are using "^^" to signify a fact type that is not derived and not stored. The other fact types are coded with "**" to indicate they are derived and stored. The derivation rules for AG and GB are the same as the value constraint in Figure 7.



Combined AG and GB derivation rule:
a,b appears in the fact type AB iff a,g appears in the fact type AG and g,b appears in fact type GB

Fig. 8. Using ^^ to illustrate transformation rules encoded in a schema

5 Conclusion

In this paper we have presented techniques that involve grouping of value instances originating from the practice of dimensional modeling. ORM is a rich modeling language that can be used to specify these modeling techniques more formally in terms of model equivalence using value constraints and derivation rules.

We have used DM techniques to motivate the generalization of the ORM VS/VC transformation to the ORM VG transformation, and we have shown that the introduction of a new notation allows us to conveniently show all fact types involved in a transformation in one schema.

We did not examine the criteria (based on symmetry of end-user query expressions) used in [10] to justify the preferred solution. If one were using a conceptual query language such as Conquer [13-14] these usability considerations may change.

References

1. Kimball, R., Ross, M.: The Data Warehouse Toolkit. John Wiley & Sons (2002)
2. Kimball, R., Ross, M., Becker, B., Mundy, J., Thornthwaite, W.: The Kimball Group Reader. John Wiley & Sons (2010)
3. Inmon, W.: Building the data warehouse, QED Technical Pub. Group (1992)
4. Halpin, T., Morgan, T.: Information Modeling and Relational Databases. Morgan Kaufmann Publishers (2008)
5. Payne, R.: MORM: a fact-oriented conceptual modeling approach to designing data warehouses and OLAP applications. MSc Thesis, University of British Columbia (2003), `https://circle.ubc.ca/handle/2429/15349`
6. Piprani, B.: Using ORM-Based Models as a Foundation for a Data Quality Firewall in an Advanced Generation Data Warehouse. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1148–1159. Springer, Heidelberg (2006)
7. Pierson, E.J., dela Cruz, N.: Using Object Role Modeling for Effective In-House Decision Support Systems. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 636–645. Springer, Heidelberg (2005)
8. Hansen, J., dela Cruz, N.: Evolution of a Dynamic Multidimensional Denormalization Meta Model Using Object Role Modeling. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1160–1169. Springer, Heidelberg (2006)
9. Kimball, R.: Help for Dimensional Modeling: Helper tables let you design and manage multivalued dimensions successfully, `http://www.kimballgroup.com/html/articles_search/articles1998/9808d05.html`
10. Kimball, R.: The Keyword Dimension: Using keywords in a data warehouse of historical letters, `http://www.kimballgroup.com/html/articles_search/articles2000/0010IE.html?TrkID=IE200010`
11. Elmasri, R., Navathe, S.: Fundamentals of Database Systems. Addison-Wesley (2007)

12. Curland, M., Halpin, T.: Model Driven Development with NORMA. In: 40th Hawaii International Conference on Systems Science (2007)
13. Bloesch, A., Halpin, T.: ConQuer: a Conceptual Query Language. In: Thalheim, B. (ed.) ER 1996. LNCS, vol. 1157, pp. 121–133. Springer, Heidelberg (1996)
14. Bloesch, A., Halpin, T.: Conceptual Queries Using ConQuer-II. In: Embley, D.W. (ed.) ER 1997. LNCS, vol. 1331, pp. 113–126. Springer, Heidelberg (1997)

A Derivation Procedure for Set-Comparison Constraints in Fact-Based Modeling

Peter Bollen

Department of Organization and Strategy,
School of Business and Economics,
Maastricht University,
P.O.Box 616,
6200 MD Maastricht, The Netherlands
p.bollen@maastrichtuniversity.nl

Abstract. In this paper we will address the conceptual schema design procedure (CSDP) in fact-based modeling. We will focus on the modeling procedure of 'cook-book' for deriving set-comparison constraints. We will give an algorithm that can be applied by an analyst in an analyst-user dialogue in which all set-comparison constraints can be derived as a result of the acceptance or rejection of real-life user examples by the domain expert.

Keywords: Conceptual Schema Design Procedure, CSDP, set-comparison constraints.

1 Introduction

In conceptual information modeling, a number of approaches have evolved over the past decades. In the mid-70's two (families of) approaches to semantic information modeling were dominant. The first approach and most popular approach was entity-relationship (ER) modeling [1]. This approach has evolved and has been extended into numerous flavours of (Extended) Entity-Relationship ((E)-E-R) approaches [2, 3]. A second approach that emerged in the mid-seventies was the fact-based approach. The earliest incarnation of this approach was the ENALIM approach [4], based on [5] that later evolved into binary NIAM [6] and subsequently into N-ary NIAM [7]. The more contemporary incarnations of these approaches are the UML class diagrams [8] for the ER family of approaches and ORM (-2) [9, 10] and CogNIAM [11, 12] for the fact-based approaches.

The main difference between the ER and fact-based modeling approaches is that the fact-based modeling approach knows only one fact-encoding construct: the fact type, in contrast to the (extended) E-R approaches in which there are (at least) two fact encoding constructs: the attribute and the relationship [13]. Another major difference between the two approaches lies in the presence of a modeling methodology or procedure in the fact-based approach [7, 14], whereas such a procedure is absent in the other approaches.

In this article we will investigate part of the modeling methodology or conceptual schema design procedure in fact-based modeling. We will propose a precise

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 329–338, 2011.
© Springer-Verlag Berlin Heidelberg 2011

specification of the conceptual schema design procedure (CSDP) [7, 9, 10] in which we will give a formal modeling procedure to derive set-comparison constraints [15] that is a further specification of steps 5 and 6 of the CSDP [9]. This specification will consist of an algorithm that can be applied by an analyst in an analyst-user dialogue leading to a complete procedure-driven derivation of all set-comparison constraints in a given Universe of Discourse (UoD).

2 The Conceptual Schema Design Procedure in Fact Based Modeling

The fact-based conceptual modeling methodology distinguishes itself from the (E)ER class of modeling approaches not only in the sense that it contains exactly one fact encoding construct instead of two, but also in the presence of a modeling methodology that can be used to create not only a syntactically correct model instance but also a semantically correct (conceptual) model for the Universe of Discourse. We will illustrate this procedure by introducing a running example from the domain of tax-collections. Earlier versions of the CSDP or ‘mini-cookbook’ can be found in [7, 16]. The most recent version of the CSDP can be found in [9] (see table 1).

Table 1. Steps in CSDP [9]

CSDP Steps	**Halpin and Morgan (2008)**
step 1	From examples to elementary facts
step 2	Draw fact types and populate
step 3	Trim schema; note basic derivations
step 4	Add uniqueness constraints and check arity of fact types
step 5	Add mandatory role constraints and check for logical derivations
step 6	Add value, subset, equality, exclusion and subtype constraints
step 7	Add other constraints and perform final checks

2.1 The Running Example: Tax Collection

In this article we will use as a running example, a simplified tax-collection universe of discourse. As a concrete example we will use the following ‘tax-collecting’ form from the (fictive) country ‘Utopia’. In the Utopian tax collection UoD citizens are requested to fill-in standardized tax collection forms. An accepted tax-form, however, has to comply to certain rules. In this article we will assume that these rules will be derived using an ‘inside-out’ approach [17] which means that we will apply the explicit steps of the Conceptual Schema Design Procedure to elicit these rules from the domain user.

UTOPIAN TAX COLLECTION FORM 2011

Social security number: 354678

Part 1

Are you married ?:

● yes ○ no

How many children do you have ?:

3

part 2

Your current work status is:

○ student
○ employee
○ unemployed

Your gross income this year was:

Fig. 1. Example 'Utopian' tax collection form

Applying step1 of the CSDP onto a significant set of tax-collection examples leads to the following fully qualified sentences:

The tax payer with social security number *354678* has a civil status with civil status code *married*.

The tax payer with social security number *354678* has an amount of children of *3*.

The tax payer with social security number *657894* has a work status with work status code *employee*.

The tax payer with social security number *657894* has a gross income of the money amount of *80 000* euros.

Once the first draft of a conceptual schema is created for a specific UoD, the analyst can elicit additional business rules from the domain user(s) by systematically confronting him/her (them) with new (combinations of) 'real-life' domain examples. The domain user only needs to confirm or reject the possibility that such a (combination of) examples can exist. The application of step 4 of the CSDP will result in the detection of the following business rules [18, 19] in the 'tax-collection' domain:

Br1: Each taxpayer has at most one civil status
Br2: Each taxpayer has at most one amount of children
Br3: Each taxpayer has at most one work status
Br4: Each taxpayer has at most one gross income money amount

The result of applying steps 1 through 4 of the CSDP is given in figure 2. The business rules that we have derived map as follows onto the roles, fact types and constraints in figure 2: *Br1* maps to the uniqueness constraint defined on role *r2* (of fact type *Ft1*). Business rule *Br2* maps to the uniqueness constraint defined on role *r4* (of fact type *Ft2*). Business rules *Br3* and *Br4* map onto the uniqueness constraints defined on roles *r5* and *r7*, respectively.

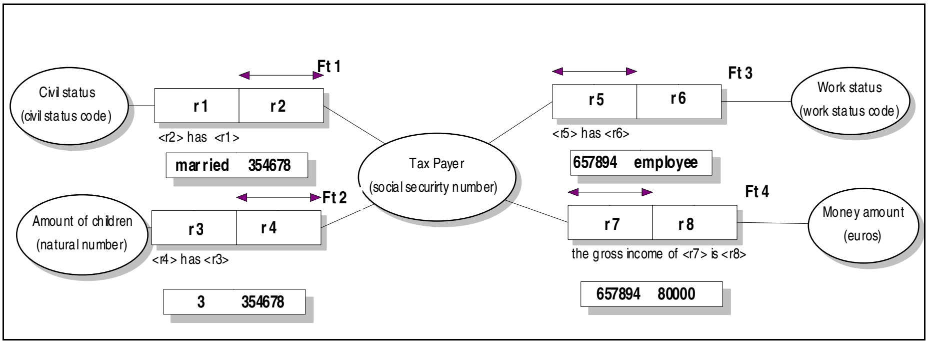


Fig. 2. Tax collection conceptual schema including uniqueness constraints

3 A Formal Derivation Procedure for Set-Comparison Constraints (CSDP Step 6)

In steps 5 and 6 of the CSDP we need to add the mandatory role, equality-, subset- and exclusion (set-comparison) constraints. In this paper we will not focus on the formal procedure for the derivation of the (disjunctive) mandatory role constraints.

A definition of set-comparison constraints, together with numerous example of these constraints can be found in section 6.4 of [9]. In this paper we will present an algorithm expressed in pseudo-code that can be used by an analyst to generate a minimal set of examples that in an analyst-user dialogue can be used to derive all set comparison constraints that govern the subject domain. The analysis is based exclusively on the acceptance or rejection of a set of (combinations) of real-life examples by the domain user in a dialogue with the (fact-based) business analyst.

Algorithm 1[1]: Set comparison constraint derivation[2].

```
BEGIN SETCOMPARISON_on_2_rolecombinations
let (R1, ....RN) and (RN+1, ....R2n)[3] be the role combinations
on which the set comparison should be performed.
Let (α1,.. αM) be a sentence instance of the fact type (FT1)
that contains roles (R1, ....RN) (M≥N)
```

[1] For a proof of the algorithm we refer to [20].

[2] We note that the algorithm assumes that objectifications or (multi-layered) nominalizations are transformed into ‘flat’ fact types (as illustrated in for example figure 3.19 of [21]).

[3] $\{R_1, ...R_N\} \neq \{R_{N+1}, ...R_{2n}\}$

Let $\beta_{N+1}, .. \beta_{2N+L}$ and $\gamma_{N+1}, .. \gamma_{2N+L}$ be sentence instances of the fact type (FT2) that contains roles $(R_{N+1}, ...R_{2n})$ (L≥0).
Furthermore, let IM:={FT1,FT2}.
Create three user examples that reflect following fact type extensions (FT1+FT2):
$EXT_1(IM)$: { $(\alpha_1, .. \alpha_M)$}
$EXT_2(IM)$: { $(\alpha_1, .. \alpha_M), (\beta_{N+1}, .. \beta_{2N+L})$ | $\alpha_1=\beta_{N+1} \wedge, .. \alpha_N= \beta_{2N}$}
$EXT_3(IM)$: { $(\alpha_1, .. \alpha_M), (\beta_{N+1}, .. \beta_{2N+L}), (\gamma_{N+1}, .. \gamma_{2N+L})$ |
$\alpha_1=\beta_{N+1} \wedge, .. \alpha_N= \beta_{2N}$}

Let the user determine which of these extensions refer to an allowed population state for the universe of discourse by showing (sets of) real-life examples that match these three extensions one at a time.

The following resulting set-comparison constraints will exist:

IF ($\exists_{Popstate1(UoD)}$ [Popstate1(UoD)= EXT_1(IM)] $\wedge$
$\exists_{Popstate2(UoD)}$ [Popstate2(UoD)= EXT_2(IM)]$\wedge$
$\neg\exists_{Popstate3(UoD)}$ [Popstate3(UoD)= EXT_3(IM)])
THEN (There is a subset constraint defined from role combination $(R_{N+1}, ...R_{2n})$ to role combination $(R_1, ...R_N)$)
ELSE IF($\neg\exists_{Popstate1(UoD)}$[Popstate1(UoD)= EXT_1(IM)]$\wedge$
$\exists_{Popstate2(UoD)}$[Popstate2(UoD)= EXT_2(IM)] $\wedge$
$\exists_{Popstate3(UoD)}$ [Popstate3(UoD)= EXT_3(IM)])
THEN(There is a subset constraint defined from role combination $(R_1,...R_N)$ to role combination $(R_{N+1},..R_{2n})$)
ELSE IF ($\neg\exists_{Popstate1(UoD)}$ [Popstate1(UoD)= EXT_1(IM)]$\wedge$
$\exists_{Popstate2(UoD)}$ [Popstate2(UoD)= EXT_2(IM)]$\wedge$
$\neg\exists_{Popstate3(UoD)}$ [Popstate3(UoD)= EXT_3(IM)])
THEN
(There is an equality constraint defined from role combination $(R_1, ...R_N)$ to role combination $(R_{N+1}, ...R_{2n})$)
ELSE IF ($\exists_{Popstate1(UoD)}$ [Popstate1(UoD)= EXT_1(IM)]
$\wedge\neg\exists_{Popstate2(UoD)}$[Popstate2(UoD)= EXT_2(IM)] $\wedge$
$\neg\exists_{Popstate3(UoD)}$ [Popstate3(UoD)= EXT_3(IM)])
THEN (There is an exclusion constraint defined from role combination $(R_1, ...R_N)$ to role combination $(R_{N+1}, ...R_{2N})$)
ENDIF
ENDIF
ENDIF
ENDIF
END

In figure 3 we have given a fact-based conceptual model plus the populations that represent the extensions 1, 2 and 3 from algorithm 1. We note that the algorithm also applies to 'nested' object types and to role combinations of nested object types and entity types.

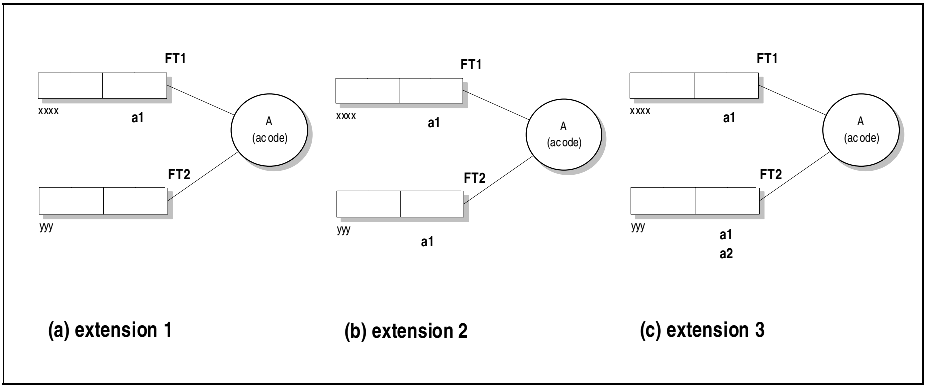


Fig. 3. Graphical fact-based presentation of extensions 1, 2 and 3 from algorithm 2

We can now summarize the algorithm as a decision-table in which a given combination of allowed existence or non-allowed existence of each of the example extensions as confirmed by the domain user in an analyst-user dialogue leads to the detection of (at most) one set-comparison constraint (see table 2). We note that for the other 4 possible outcomes of the algorithm no set-comparison constraints will be derived.

Table 2. Decision logic from the set-comparison constraint derivation algorithm 2

	1	**2**	**3**	**4**
EXT1	Allowed	Not Allowed	Not Allowed	Allowed
EXT2	Allowed	Allowed	Allowed	Not Allowed
EXT3	Not Allowed	Allowed	Not Allowed	Not Allowed
Constraint type	Subset1 (FT2->FT1)	Subset2 (FT1->FT2)	equality	exclusion

The scope of Algorithm 1 is one pair of role(s) (combinations) in which in principle the same entity types (or nested object types or a combination of entity types and nested object types) are involved. In our tax collection example roles r2, r4, r5 and r7 are the roles played by the entity type *tax payer*. Hence, for each combination of two roles[4] that potentially can have set-comparison constraints attached to them we must apply algorithm 1.

In this example of a conceptual schema we, therefore, need to apply algorithm 1 for the following pairs of roles: (r2, r4); (r2, r5); (r2, r7); (r4, r5); (r4, r7); (r5, r7). To

[4] Note that the number of combinations of two roles that have to be analyzed in this example are determined by the binomial coefficient which in this case equals: 4!/(2!*2!) = 6.

illustrate the derivation algorithm we will show the application of the algorithm for one out of six pairs of roles (r2, r4) in the form of a constructed analyst-user dialogue in which 3 tangible domain examples are constructed that match the extensions 1, 2 and 3.

We will consider the potential set-comparison constraint that we will derive for the roles r2 and r4. Extension 1 is supposed to be a sole instance of fact type 1: (married, 354678). The expression of this fact instance as a tangible user example is given in figure 4, the verbalization of this example is:

The civil status for tax payer *354678* is *married*

If we confront a domain user with this example and ask him/her to state whether this is an allowed example of communication or not, the answer is; yes, this is an allowed example of communication. We can now move on to the creation of a tangible example that embodies the example facts that are contained in extension 2 (as defined in algorithm 1). Extension 2 represents an instance of fact type 1: (married, 354678) plus an instance of fact type 2 (3, 354678), the verbalization of this example is:

The civil status for tax payer *354678* is *married*
Tax payer *354678* has *3* children

If we confront a (different) domain user with this example and ask him/her to state whether this is an allowed example of communication or not, the answer is: yes, this is an allowed example of communication. We can now verbalize the business rule that is exemplified by this acceptance of the example by the user as follows:

Br5: A taxpayer that has a marital status can have children.

Extension 3 represents an instance of fact type 1: (married, 354678) plus two instances of fact type 2: (3, 354678) and (3, 784567). The verbalization of this example is:

The civil status for tax payer *354678* is *married*
Tax payer *354678* has *3* children
Tax payer *784567* has *3* children

If we confront a (different) domain user with this example and ask him/her to state whether this is an allowed example of communication or not, the answer is: NO, this is not an allowed example of communication. The reason that it is not an allowed example is that if the number of children are listed, a marital status must also be given (note that this is not the case for taxpayer 784567). We can now verbalize the business rule that is exemplified by this acceptance of the example by the user as follows:

Br6: A taxpayer that has children must have a marital status.

When we now apply the decision table (2) in terms of algorithm 1 we yield the following set comparison constraint that exists.

`There is a subset constraint defined from role combination(`R_{N+1}`, …`R_{2n}`)to role combination (`R_1`, .`R_N`)`

Which in this situation means a subset constraint that is defined from role *r2* to role *r4* (see figure 4) exists. We will show the final result of applying algorithm 1 on all 6 role combinations in the following set of (additional) business rules to the fact types and uniqueness constraints in figure 2.

Br5: A taxpayer that has a marital status can have children.
Br6: A taxpayer that has children must have a marital status.
Br7: A taxpayer that has a marital status cannot have a work status
Br8: A taxpayer that has a marital status cannot have a gross income
Br9: A taxpayer that has a gross income must have a work status
Br10: A taxpayer that has a work status must have a gross income

The accompanying resulting conceptual schema including uniqueness and set comparison constraints is presented in figure 4. The business rules that we have derived map as follows onto the fact types and constraints in figure 4: *Br5* and *Br6* map to subset constraint *c1*, business rule *Br7* maps onto exclusion constraint *c2*, business rule *Br8* maps onto exclusion constraints *c4*. Finally, business rules *Br9* and *Br10* map onto equality constraint *c3*.

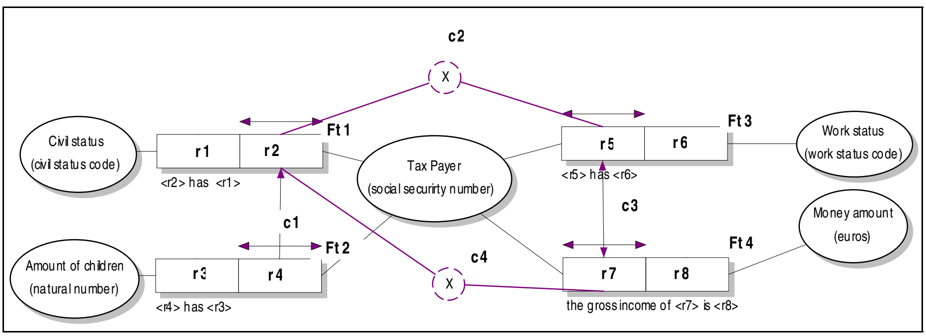


Fig. 4. Conceptual schema including uniqueness- and set-comparison constraints

We have now shown how a formal procedure will help us in extracting the domain-knowledge in the form of set-comparison constraints that can be phrased as semantic business rules by the domain user.

4 Conclusion

In this paper we have shown how the fact-based modeling methodology can help a business analyst to explicitly derive all set-comparison constraints in an analyst-user dialogue, by exclusively confronting the domain expert with ‘real-life’ examples of communication. The specific format of these examples is defined in algorithm 1 which is a formalization of (parts of) steps 5 and 6 of the Conceptual Schema Design Procedure (CSDP). The only domain input for this procedure is the result of the acceptance or rejection of real-life user examples by the domain expert.

As a future research project we will investigate how the algorithm in this paper can be used to derive the (disjunctive) mandatory role constraints that govern the universe of discourse. The objective is to extend the current procedure that was introduced in this paper in tandem with a mandatory role to set-comparison constraint mapping schema, resulting in a fully procedure-based derivation of (disjunctive) mandatory role constraints (see [20]).

References

1. Chen, P.: The Entity-Relationship Model: Toward a Unified View. ACM Transactions on Database Systems 1(1), 9–36 (1976)
2. Teorey, T., Yang, D., Fry, J.: A logical design methodology for relational databases using the extended E-R model. ACM Computing Surveys 18(2), 197–222 (1986)
3. Hoffer, J., Prescott, M., McFadden, F.: Modern database management, 7th edn. Pearson/Prentice-Hall (2005)
4. Nijssen, G.: On the gross architecture for the next generation database management systems. In: Information Processing. IFIP (1977)
5. Falkenberg, E.: Significations: the key to unify data base management. Information Systems 2, 19–28 (1976)
6. Verheijen, G., van Bekkum, J.: NIAM: An Information Analysis method. In: IFIP TC-8 CRIS-I Conference. North-Holland, Amsterdam (1982)
7. Nijssen, G., Halpin, T.: Conceptual schema and relational database design: a fact oriented approach. Prentice-hall, New-York (1989)
8. Booch, G., Rumbaugh, J., Jacobson, I.: Unified Modeling Language User Guide, 2nd edn. The Addison-Wesley Object Technology Series. Addison-Wesley Professional (2005)
9. Halpin, T., Morgan, T.: Information Modeling and Relational Databases; from conceptual analysis to logical design, 2nd edn. Morgan-Kaufman, San-Francisco (2008)
10. Halpin, T.: Information Modeling and Relational Databases; from conceptual analysis to logical design. Morgan Kaufmann, San Francisco (2001)
11. Nijssen, M., Lemmens, I.: Verbalization for Business Rules and Two Flavors of Verbalization for Fact Examples. In: Chung, S., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 760–769. Springer, Heidelberg (2008)
12. Nijssen, M., Lemmens, I., Mak, R.: Fact-Orientation Applied to Develop a Flexible Employment Benefits System. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 745–756. Springer, Heidelberg (2009)
13. Bollen, P.: How to overcome pitfalls of (E)ER and UML in Knowledge Management education. International Journal of Teaching and Case Studies 1(3), 200–223 (2008)
14. Nijssen, G.: Grondslagen van Bestuurlijke Informatiesystemen. Nijssen Adviesbureau voor Informatica, Slenaken (1989)
15. Leung, C., Nijssen, G.: From a NIAM conceptual schema onto the optimal SQL relational database schema. Australian Computer Journal 19(2), 69–75 (1987)
16. Nijssen, G.: Mini cookbook to prepare a conceptual schema or Conceptualization Procedure in fourteen major steps. University of Queensland, Australia (1982), `http://www.sjirnijssen.eu/documents.aspx`
17. Bollen, P.: A Fact-Based Meta Model for Standardization Documents. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 464–473. Springer, Heidelberg (2010)

18. Bollen, P.: SBVR: A Fact-Oriented OMG Standard. In: Chung, S., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 718–727. Springer, Heidelberg (2008)
19. Bollen, P.: The Orchestration of Fact-orientation and SBVR. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) Enterprise, Business-Process and Information Systems Modeling. LNBIP, vol. 29, pp. 302–312. Springer, Heidelberg (2009)
20. Bollen, P.: A derivation procedure for mandatory role and set-comparison constraints in fact-based modeling, in Meteor Research Memorandum RM/11/036, Maastricht University, School of Business and Economics: Maastricht (2011), http://edocs.ub.unimaas.nl/loader/file.asp?id=1617
21. Bakema, G., Zwart, J.-P., van der Lek, H.: Fully Communication Oriented Information Modeling (FCO-IM), Academic Service (2002)

A Dialogue Game Prototype for FCO-IM

Stijn Hoppenbrouwers[1], Albert Pratama Jemmi Oscar[2],
Dimas Listianto Adisuryo[2], and Jan Pieter Zwart[2]

[1] Radboud University Nijmegen,
Institute for Computing and Information Sciences,
P.O. Box 9010, 6500 GL, Nijmegen, The Netherlands
stijnh@cs.ru.nl
[2] HAN University of Applied Sciences,
Academy of Communication and Information Technology,
Ruitenberglaan 26, 6826 CC, Arnhem, The Netherlands
albert.pratamajemmioscar@gmail.com,
quatrosofia@yahoo.com,
JanPieter.Zwart@han.nl

Abstract. We report on the development of a prototype dialogue game for FCO-IM. Dialogue games are systems for executing controlled conversations. As part of an ongoing effort to analyze and support collaborative modeling processes by means of dialogue games, we created a dialogue game prototype for essential parts of the FCO-IM modeling process. The project was exploratory in nature, but results suggest that eventually it should be possible to develop dialogue games that could effectively support fact-based conceptual modeling efforts. In addition, work on such games enhances our insights in the FCO-IM modeling process, and leads to improvements and extensions of its operational guidelines.

Keywords: Fact Oriented Modeling, conceptual modeling, FCO-IM, collaborative modeling, dialogue game, process of modeling.

1 Introduction

Compared to mainstream conceptual modeling approaches like UML, the various approaches to Fact Oriented Modeling (FOM) have always included a relatively well defined process or procedure for executing their modeling practices. For example, in ORM [1] this is known as the CSDP (Conceptual Schema Design Procedure). This paper focuses on another flavor of FOM: FCO-IM [2], which includes an even more elaborate modeling procedure.

In the past, initial steps have been taken to better understand, and ultimately better support, the process of FOM [3-5]. A driver behind these efforts has been the conviction that better support is needed for the process of conceiving and expressing conceptual models *beyond* the use of diagram editors like NORMA, CaseTalk, or Graphity. This actual process of modeling is at least as essential to the final product as is the modeling language. Also, the practice of using FOM in system development and business analysis processes would benefit from more accessible, easier

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 339–349, 2011.
© Springer-Verlag Berlin Heidelberg 2011

procedures requiring less training and experience for the production of good quality models [6].

This line of research now revisits FOM, from which it originated. It brings along accumulated knowledge from fields like Method Engineering [7], Collaborative Modeling [8] and Knowledge Engineering [9].

Chiefly, this paper reports on a Master's Thesis project at HAN University of Applied Sciences, Arnhem, the Netherlands. Radboud University Nijmegen participated as problem owner and provided input and high level requirements for a prototype system embodying some key parts of the existing FCO-IM modeling procedure. This included an elaborate analysis of this procedure within the framework of 'Dialogue Games', as will be explained below, as well as 'proof of concept' design, implementation, and evaluation of the system.

2 Theoretical Retrospect: Modeling as a Dialogue Game

Though the long term goal of the 'methods as dialogue games' line of research is to develop improved support systems for processes of system modeling, analysis, and development (within the Design Science paradigm), it also involves more fundamental work on what modeling is [10] (especially in a collaborative setting), what it is *for*, and what is asked of people when they model. From the outset, we have taken a communication perspective on these matters. Though the *products* (artifacts) resulting from modeling are firmly based on formalisms and formal representations (e.g. predicate logic, Petri nets), the activities of *eliciting*, *conceptualizing*, *expressing*, and *discussing* or even *negotiating*, and *validating* such artifacts constitute a step-by-step process that is rooted in human cognition, language, and social interaction. The challenge is to bring together the world of thinking using formal systems and the world of 'regular' human experience, thinking and communication [11]. This challenge is particularly hard (and interesting) if people are involved that have little or no training in thinking formally. The FOM tradition (contrary to many other approaches in conceptual modeling) has always included such considerations (for example, by using fact statements by domain experts as basis for further abstraction, and by deploying natural language verbalization mechanisms besides more technical diagrams), but has hardly considered them at a theoretical level focusing on thought processes of and interaction between individuals. We set out to remedy this.

We initially aimed for understanding, capturing and providing *strategies* for modeling [4]. We found that to understand strategies, we first need to know what *goals* they work towards, in what *situations* and under which *constraints*. Also, the process of modeling cannot be fruitfully captured by a detailed step-by-step 'recipe', but rather concerns a set of goal-based constraints on expressions and interactions that allows for considerable freedom in the execution of a set of necessary steps. A declarative, rule-based way of describing goals and constraints proved better suited for capturing and guiding modeling processes than an imperative one using, for example, flow charts [6]. The next step was the realization that all this pointed towards the analysis and shaping of modeling processes as *games*, which are by nature *interactive systems* [7].

Next, the combination of basic discourse theory and study of interactions during modeling sessions led to the development of the RIM (Rules, Interactions, Models) framework [12]. This takes the view that **M**odels are *sets of propositions* put forward, evolved and selected in *conversations* by means of ***I****nteractions* like 'propose', 'agree', 'disagree', 'accept', 'reject', which are governed and constrained by various types of ***R****ules* like grammar, content, interaction and procedure rules [8]. Many secondary aspects are integrated in the modeling process besides the core activity of 'putting together the model', including planning and even discussion of modeling concepts to be used (i.e. specifics of language and notation) [12].

The RIM framework and the game approach came together in the notion of *dialog games*: a theoretical concept from Argumentation Theory [13]. This theoretical notion was operationalized by Andrew Ravenscroft and his team, resulting in the InterLoc system for setting up and playing dialogue games for generic discussion [14]. It is the basic interface and interaction mechanisms of this system that inspired our prototype dialogue game for FCO-IM.

The InterLoc approach includes the explicit structuring of "moves" in a discussion by making people choose moves like "asking", "stating" and "refuting" in a chatbox-like environment. The main mechanism to operationalize this is the use of "openers". During the chat dialogue, participants (or players) have to select an opener for each line they enter. For example, generic openers are "I'd like to state the following: …", "I have a question about this: …", or "That's total rubbish, because …". Setting such openers influences the possible structure and tone of the dialogue, and indicates to players what moves they can make. To extend this approach to a conceptual modeling context, the conceptual constraints associated with the use of a modeling language (syntax) had to be added. This was simply done by including such content into openers, e.g. "I propose the following fact type name: …". The InterLoc approach also renders a log of all interactions that lead to a model. The model can be directly derived from the conversation (at every point in time) by means of gathering all accepted conclusions occurring in the log.

In addition, it helps greatly if an up-to-date diagrammatic version is continuously shown to all players. We first successfully did this in the context of Group Model Building, a collaborative modeling discipline within Systems Dynamics modeling [15]. Based on these experiences, the current project was initiated: to apply a dialogue game approach to FCO-IM, in an explorative attempt to frame the FOM process as a dialogue game.

We are aware that executing a conversation for modeling strictly through the mechanism of a structured textual chat can hardly be expected to be user friendly and efficient in a real modeling context. Therefore, we have from the beginning envisaged the augmentation of the *deep interface* (i.e. the chatbox and log) with a *surface interface*: a collection of form-like and graphical interfaces as well as visualizations providing efficient and user friendly short-cuts for entering the verbal dialogue. The projection of a current diagram is the most basic form of such a surface interface, but more advanced and more interactive mechanisms are expected to be of great benefit. Using such surface mechanisms will still result in the generation of deep interface log entries for the ongoing dialog, while also allowing for direct conversational entries in the log through the structured chat interface. In the next section, we present an outline of our prototype dialogue game for FCO-IM.

3 The Prototype: Analysis, Design, Evaluation

The goal of our project was to create a prototype of a dialogue game for FCO-IM, or at least for some essential parts of the procedure. Though dialogue games inherently involve multi-player interactions, organizational and technical restrictions within the project made it impossible to develop the prototype on a distributed architecture. Instead, a single client was created that was used by all players taking turns. While this is not a realistic solution for practice, it served well to explore and test the principal interaction mechanisms.

The two usual roles in FCO-IM were discerned in the dialogue game: the *domain expert* and the *information analyst*. In the initial dialogue game prototype created for the Group Model Building approach (see Section 2), the role of *facilitator* was explicitly used. For the FCO-IM game, this role was less prominently included as ‘Game Master’: the Game Master just controls the focus of conversation, switching between a range of sub-games or ‘modes’ as required.

Three main parts in the FCO-IM modeling procedure were distinguished in the prototype, in line with standard FCO-IM modeling:

1. Stating elementary facts [2, Ch. 2.3]: the basis of FCO-IM modeling;
2. Classification and Qualification [2, Ch. 2.4] of the facts given in 1;
3. Constraint determination applied to (populations of) the fact types created in 2. Because of time restrictions of the project, only the assignment of uniqueness constraints [2, Ch. 3.2] were covered by our prototype.

These three parts were chosen from the FCO-IM operational procedure [2] because they cover the spectrum from exhaustively prescribed steps (for part 3, [2, 18]), via partially prescribed steps (part 2) to only tentatively heuristically described steps (1).

The next step in the prototype development was to analyze the existing FCO-IM procedure in terms of Focused Conceptualizations (or FoCons) [16]. The main thrust of a FoCon analysis is to identify the various conceptual foci as they occur throughout a modeling session. FoCons concern the operational focus in part of a conversation for modeling, often rendering *intermediate products*. They are often sub-foci of those directly related to the *end product* of the modeling effort (in our case, an FCO-IM model). Though it is possible that such foci and their related *game modes* are applied in a preset sequence, observations of collaborative modeling sessions of various types have shown that foci are often switched reactively, in a more ad hoc fashion [12]. It should therefore be possible to either apply foci in an orderly, systematic fashion, but also to (re)visit some focus/mode unexpectedly, as required by the current dynamics of conceptualization and discussion.

Below, we provide an overview of the FoCons identified, though lack of space unfortunately prevents us from giving the full detail of the foci and the set question types and answer types involved.

Stating Elementary Facts. The sub game of ‘stating elementary facts’ concerns a highly creative activity that is hard to cover by a fixed procedure. As a focused conceptualization, it requires the domain expert to produce elementary fact statements (the ‘output’ of the FoCon) based on representative documentation (the ‘input’) reflecting actual communication as takes place in the domain. In addition, all fact statements need to be *structured* in a particular format: elementary facts. This FoCon has no sub-modes, i.e. is covered by only one mode, with 7 openers (see section 2 for an explanation of 'openers').

Classification and Qualification. This FoCon follows a more algorithm-like procedure than the previous one, which provides its input: elementary fact statements. Its output is a set of 'meaningful classes of facts and objects', with the structure restriction that the fact types thus stated are correctly classified and qualified, i.e. are phrased in terms of correct object types (with their proper identification) and/or label types. The FoCon is covered by four separate conceptualization modes, with 10 to 15 openers; however, quite a few of those openers (in particular the generic ones like "I agree, because …") were used in more than one, if not all, modes.

Assigning Uniqueness Constraints (UCs). This third main FoCon involves a strict procedure which takes as input a fact type expression (provided through the second main FoCon) as well as its population (provided through the first main FoCon, but possibly also purpose-created). The output is a set of 'uniqueness constraints' correctly assigned to the roles of the fact type. The FoCon roughly involves presenting the domain expert with two fact statements populating the same fact type and answering a question about this configuration: "can these two fact statements hold (at the same time)?". The actual procedure is more complex, involving various iterations of the questions and covering various sub-items within a fact type [18]. No fewer than eight different modes of conceptualization were identified, with 2-9 openers.

Importantly, not only openers were provided but also rules (to be executed by the game master) concerning when to switch from one mode to another. However, note that such a switch can in principle be made at any point (possibly even at the initiative of a player), though usually one will want to finish a single sub-activity in order to avoid confusion and keep progress transparent.

Table 1. Some examples of openers used in various sub-modes of the 'assign UC' mode

Role	**Interaction type**	**Opener**
Game master	Directive	Start mode for UC determination for fact type
Inf. analyst	Proposition	Consider the following fact statements: …
Inf. analyst	Question	Can these statements occur together?
Inf. analyst	Proposition	There is a UC on role(s) … of the fact statement
Domain exp.	Answer	No.
Domain exp.	Answer	Yes.
Game master, Domain exp., Inf. analyst	Question (generic)	I have a question: …
Domain exp. Inf. analyst	Disagreement (generic)	I disagree, because …

To keep a clear overview of openers to choose from at a certain point in the game, but also to impose focus and maintain a certain level of control, openers available for use at a certain moment are restricted by the role of the player who intends to make the dialog move, and the mode that is currently active. As a limited example, Tab. 1 shows some typical openers from various phases of the 'assign uniqueness constraints' FoCon, including the roles to use them and the interaction type they fall under. The bottom two examples are generic, while the others are more specific for one or more (UC related) modes.

Let us now turn to the way this design was implemented in a prototype dialog game system. We will only consider the main functionality of the system here, i.e. the actual dialog game. For reasons of space, we discard functionality concerning 'project and discussion management' and 'user and game role management', which is fairly straightforward.

The prototype was developed using two components: Microsoft SQL Server 2008 as database management system, and Microsoft Visual Studio with the visual basic dot net (VB.Net) language to develop the windows application. The connection between the windows application and the database was done by using an ODBC connection.

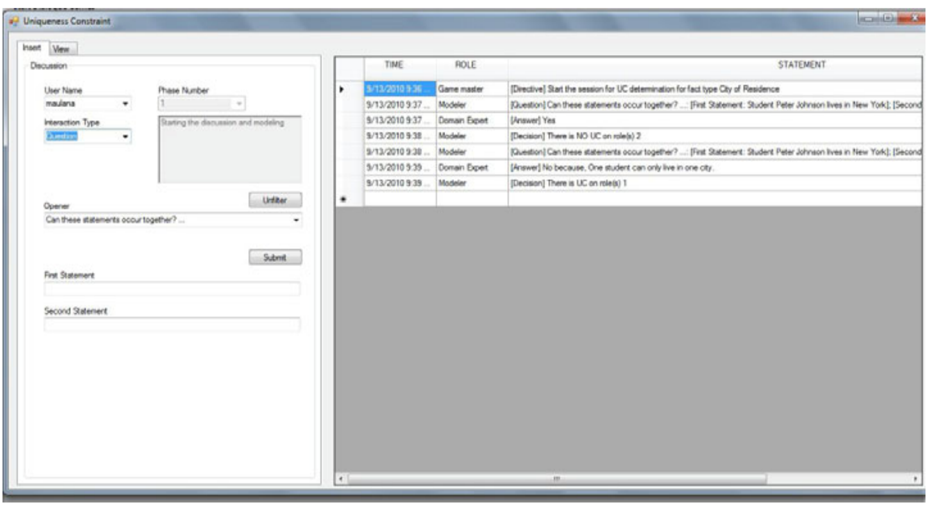


Fig. 1. UC determination Deep Interface, the main window of the prototype

The dialogue game deep interface consists of three forms, one for each main FoCon or sub-game in the FCO-IM procedure. Each form consists of two parts. The left part is used to enter the conversation (for each user, switching between them). The right part is used to show the conversation log of the game. Fig. 1 shows the deep interface (see section 2) of a UC determination dialogue game; the 'surface interface', introducing more advanced and efficient interaction and visualisation, was not included in this prototype.

The left part of the form consists of the population of User Name, Mode Number (named Phase Number in the screenshot), Mode Description, Interaction Type, and Opener. Mode Number is used to filter what interaction types and openers are available for the current mode in the discussion. For each user name and its related discussion role, there can be a unique interaction type on different mode numbers. The mode number can only be changed by the game master since she is the one who guides the discussion and determines its focus. However, if required, the filter can be temporarily removed, allowing players to use all openers of all modes in the game.

For each FoCon, there are several groups of openers which require different types of input fields used to insert the free text part(s) following the openers. These free text parts are used to show known elements of the model (or derivatives thereof) required in some specific statements and questions. For example, they hold two fact statements that are to be compared. In the prototype, the elements have to be entered manually; in future versions, such elements could be imported straight from a model repository. If there is one free text part, one input field is enabled; if there are two free text parts, two input fields are enabled, and so on.

All inserted conversations are logged (showing a timestamp and the user entering the entry) and can be viewed through the data grid on the left side of the form. In Table 2, a short but representative part of a test dialogue for 'UC Determination for fact type' is shown.

Table 2. A part of the log for a session in the UC Determination mode

2010-09-13 23:53:29.563	Information Analyst	[Question] Can these statements occur together? [First Statement: Student Peter Johnson lives in New York]; [Second Statement: Student John Peterson lives in New York]	UC-D-FactType
2010-09-13 23:53:40.633	Domain Expert	[Answer] Yes	UC-D-FactType
2010-09-13 23:54:00.947	Information Analyst	[Decision] There is no UC on role 2	UC-D-FactType
2010-09-13 23:54:32.667	Information Analyst	[Question] Can these statements occur together? [First Statement: Student Peter Johnson lives in New York]; [Second Statement: Student Peter Johnson lives in Houston]	UC-D-FactType
2010-09-13 23:55:01.990	Domain Expert	[Answer] No, because one only can live in one city	UC-D-FactType
2010-09-13 23:55:17.980	Information Analyst	[Decision] There is a UC on role 1	UC-D-FactType
2010-09-13 23:55:29.443	Game Master	[Directive] The session for UC determination for this fact type is finished	UC-D-FactType

Following its initial implementation, the prototype was tested in an elaborate session with three students (not otherwise involved in the project but familiar with FCO-IM) who took turns in playing the ‘information analyst’ and ‘domain expert’ (switching roles after each main FoCon ended). One of the researchers acted as ‘game master’. After the test, improvements were carried through. Next, the system was tested one final time in another, more limited session not involving ‘external’ people.

The initial, elaborate game session in particular aimed to capture the response of the participants when they played the dialogue game for FCO-IM, and also to get some feedback from the participants in order to refine the dialogue game system as such. The complete session took two hours in total. It led to the following relevant improvements of the system (we disregard technical problems and debugging):

1. The openers were refined by replacing the initial ‘dotted blanks’ in statements (as still shown in Table 1) with more explicit indications: [Expression 1], [Expression 2], etc. This improved clarity for players.
2. The users had no trouble using the openers in the Fact Statement and UC Determination modes. However, the Classification and Qualification modes were problematic in their current form. Questions that can be asked by the information analyst are more open than in the other phases. The modeller could use several openers provided in the prototype in common cases of classification and qualification. However, if a special case was encountered, such as semantic equivalence, the questions to be asked by the modeler are highly dependent on the context of the case itself. We conclude that this mode needs to be reconsidered and redesigned.
3. Various terms used in openers (even quite basic ones like “class”, “classification”, and “qualification”) were too difficult for the players to be understandable and usable in the game. It seems recommendable to either rephrase them in more accessible language or form, or alternatively to much more elaborately instruct the players (probably in advance). We lean towards the first option.
4. The users found it difficult to follow the flow of the game (switching between modes), since users (in particular when acting as ‘information analyst’) were insufficiently familiar with the dialogue game modes and their underlying procedures. A possible solution might be for the game master to more actively engage in facilitating the process (not unlike in the GMB dialog game; in fact, perhaps the roles of information analyst and game master could be merged). However, simplifying the game is another viable option (in line with 3. above). Even more than in the case of 3., very elaborate instruction of the players seems undesirable, since the game is intended to, eventually, support the modelling process, not vice versa.

4 Conclusions and Further Research

We are quite aware that *in its current form*, the prototype comes nowhere near providing realistic support for FCO-IM modeling. In fact, we did not expect it would. We are merely exploring principles of shaping and structuring the act of modeling. However, we managed to take a first, substantial step in taking the existing FCO-IM modeling procedure and transforming it into something that is the basis for a 'collaborative system' in the sense of [17]. Expected advantages of such an approach (when successful) are:

1. Complete records of the conceptualization of the model, including rejected options and modifications, and also (importantly) all communication *about* the model (including rationales for modeling decisions). This is useful for later reference, but also for systematic analysis of the interactive process of modeling in view of improvement of the process. In this sense, dialog games as discussed here are perhaps mostly a means for developing more evolved approaches rather than a definite solution.
2. The use of foci or 'modes' in a framework for enactment of conceptualization reflects the natural way of *thinking* (not just 'model editing') that occurs in most if not all collaborative conceptual modeling efforts. The way the modes are managed and deployed makes it possible to strike a balance between providing structure and allowing for a degree of freedom (ad hoc initiative and liberal iteration), as required by the nature of collaborative modeling.
3. The dialog frame can in principle be used as a basis for much more advanced interactive systems, combining accumulated findings of various experimental games into a generic, deep interface system that can be parameterized, and augmented with elaborate surface interface functionality. Importantly, the deep structure interface can thus be combined with existing surface structure functionalities like advanced editors, verbalizations, user friendly forms, etc.

Looking at our first prototype, we conclude that simplification is the main goal for improvement. Usage of specialized FCO-IM terms in focus questions (and set answers) should be minimized. Some parts of the FCO-IM modeling practice might be simplified, not just by reducing the use of specialized terms involved, but also by defining even smaller, more understandable and focused steps and providing clear-cut and 'natural' examples during the game. Other parts might be supplemented with hints, heuristics or more detailed guidelines. The current prototype was only a first attempt to capture the *existing* FCO-IM procedure in a dialogue game. Simplification certainly is a challenge, which does not so much concern the general setup of the dialog game system with its modes and openers, but the FCO-IM modeling procedure as such. Indeed, we intend the dialog game framework to be a platform for studying and improving procedures for conceptual modeling.

This is not to say that the general setup should not also be improved. Main limitations of the prototype (and directions for improvement) include the following:

- The system is not actively and automatically linked with a real model repository. Doing so would greatly improve the functionality and efficiency of the game.
- In the InterLoc system, a threading mechanism is available, providing very helpful additional structure in the dialog log and greatly aiding navigation and analysis. Since it has been observed that players indeed like to refer back to previous statements whilst playing the game, threading and other navigation aids would be a substantial improvement.
- Obviously, it is desirable to generate verbalizations and visualizations based on the model elements gathered in the repository. However, such items should be made available only if they are really helpful. For example, overly complex

diagrams should not be shown to players who have no grasp of them, since this would only confuse them and create information overload. *Focus* remains important.

- Even more obviously, any realistic version of the dialog game requires a distributed, multi-client implementation – as is already present in the InterLoc system.
- Currently, the game master has to know everything about the modes and their required succession. Intelligent aids (based on formalized and decidable rules) may help a game master to master the game, lowering expertise needed to facilitate a game session.

Given various non-trivial aspects of the conceptualization of properly formalized models, there will doubtlessly be limits to making such a process accessible and 'disintermediated'. Nevertheless, we do believe that considerable progress can be made in this respect. Clear and effective interplay between the differentiated roles involved, and adjusting the games to the skills and knowledge of typical people playing these roles, may seriously improve the effective organization of interaction between participants in FCO-IM and other conceptual modeling sessions.

References

1. Halpin, T.A.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann, New York (2008)
2. Bakema, G., Zwart, J.P., van der Lek, H.: Fully Communication Oriented Information Modeling (FCO-IM). FCO-IM Consultancy, Arnhem (2002), Downloadable from `http://www.fco-im.nl`
3. Hoppenbrouwers, S.J.B.A(S.), Lindeman, L(L.), Proper, H.A(E.): Capturing Modeling Processes – Towards the MoDial Modeling Laboratory. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1242–1252. Springer, Heidelberg (2006)
4. Hoppenbrouwers, S., Proper, H.A.E., van der Weide, T.P.: Towards Explicit Strategies for Modeling. In: Halpin, T.A., Siau, K., Krogstie, J. (eds.) Proceedings of the Workshop on Evaluating Modeling Methods for Systems Analysis and Design (EMMSAD 2005), Held in Conjunctiun with the 17th Conference on Advanced Information Systems (CAiSE 2005), pp. 485–492. FEUP, Porto (2005)
5. van Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, H.A(E.), van der Weide, T.P.: Exploring Modelling Strategies in a Meta-modelling Context. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1128–1137. Springer, Heidelberg (2006)
6. van Bommel, P., et al.: Concepts and Strategies for Quality of Modeling. In: Halpin, T., Krogstie, J., Proper, H.A.E. (eds.) Innovations in Information Systems Modelling, Methods and Best Practices, pp. 167–189. IGI Global Publishing, Hershey (2008)
7. Hoppenbrouwers, S., Bommel, P.v., Järvinen, A.: Method Engineering as Game Design: an Emerging HCI Perspective on Methods and CASE Tools. In: Halpin, T., Proper, H.A., Krogstie, J. (eds.) Proceedings of the 13th International Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2008), Montpellier, France, CEUR digital proeceedings (2008)

8. Hoppenbrouwers, S., Weigand, H., Rouwette, E.: Setting Rules of Play for Collaborative Modeling. International Journal of e-Collaboration (IJeC) 5(4), 37–52 (2009)
9. Hoppenbrouwers, S., Schotten, B., Lucas, P.J.F.: Towards Games for Knowledge Acquisition and Modeling. International Journal of Gaming and Computer-Mediated Simulations, Special Issue on AI and Games 2(4), 48–66 (2010)
10. Hoppenbrouwers, S., Proper, H.A., van der Weide, T.P.: A Fundamental View on the Process of Conceptual Modeling. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 128–143. Springer, Heidelberg (2005)
11. Hoppenbrouwers, S.: Community-based ICT Development as a Multi-Player Game. In: Benoit-Barné, C., et al. (eds.) What is an Organization? Materiality, Agency, and Discourse: a Tribute to the Work of James R. Taylor, Dept. of Organizational Communication, University of Montreal, Montreal (2008)
12. Ssebuggwawo, D., Hoppenbrouwers, S., Proper, E.: Interactions, Goals and Rules in a Collaborative Modelling Session. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 54–68. Springer, Heidelberg (2009)
13. Eemeren, F.H.v., et al.: Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments. Lawrence Erlbaum Associates, Mahwah (1996)
14. Ravenscroft, A., McAlister, S.: Designing interaction as a dialogue game: Linking social and conceptual dimensions of the learning process. In: Juwah, C. (ed.) Interactions in Online Education: Implications for Theory and Practice, pp. 73–90. Routledge, New York (2006)
15. Hoppenbrouwers, S., Rouwette, E.: A Dialogue Game for Analysing Group Model Building: Framing Collaborative Modelling and its Facilitation. International Journal of Organisational Design and Engineering, Special Issue on Collaborative Modeling (to appear, 2011)
16. Hoppenbrouwers, S., Wilmont, I.: Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J., et al. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 190–204. Springer, Heidelberg (2010)
17. Kolfschoten, G., et al.: A conceptual foundation of the thinkLet concept for Collaboration Engineering. International Journal of Human-Computer Studies 64, 611–621 (2006)
18. Zwart, J.P., Bakema, G.: Advances in FCO-IM (2): A Shorter Algorithm for Determining Intra Fact Type Uniqueness Constraints. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 719–728. Springer, Heidelberg (2007)

Directly Applied ORM Constraints for Validating and Verifying Semantic Decision Tables

Yan Tang

VUB STARLab,
10G731, Vrije Universiteit Brussel,
Pleinlaan 2, 1050 Elesene,
Brussels, Belgium
yan.tang@vub.ac.be

Abstract. A semantic decision table (SDT) is a decision table properly annotated with domain ontologies. In this paper, we focus on validation and verification issues (V&V) for SDT. We use Semantic Decision Rule Language (SDRule-L), which is an extension to the Object-Role Modeling language (ORM/ORM2) for modeling SDT commitments. With these commitments, business rule modelers can discover invalid decision rules. In this paper, we focus on how to use six SDRule-L constraints, which are directly brought from ORM/ORM2. Our approach focuses on detecting invalid rules within one table.

Keywords: Object-Role Modeling, ORM/ORM2, Validation and Verification, V&V, semantic decision table.

1 Introduction

Decision table is a means to model business rules in a tabular format. It has been used for decades in many domains, especially in the business domain seeing that it has the advantage of clear visualization of decision rules for non-technical business people.

As an extension to decision tables, semantic decision tables (SDT, [12]) use modern ontology technologies to model meta-rules and specify hidden semantics of decision tables. In this paper, we will focus on the validation and verification issue (V&V) for SDT in order to ensure the quality of the modeled decision rules.

A decision table may depend on other tables [2]. Sub-table and meta-table are the two examples. Like decision tables, SDTs can be layered and crossly linked. In this paper, we focus on dealing with the dependencies *within one* table instead of across tables. We use Semantic Decision Rule Language (SDRule-L, [13]) to model the constraints and axioms in SDTs, which we call SDT commitments. SDRule-L is an extension to ORM/ORM2 [1]. In this paper, we focus on the SDRule-L constraints that are directly brought from ORM/ORM2 and demonstrate how these constraints can be applied for SDT V&V. They are the constraints of *value*, *uniqueness*, *mandatory*, *frequency*, *exclusive-or* and *subtyping*.

The paper is organized as follows. Sec. 2contains the paper background. Our related work is illustrated in Sec. 3. We discuss how ORM/ORM2 can be directly applied for SDT V&V in Sec. 4. We conclude and discuss our future work in Sec. 5.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 350–359, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Background

A *decision table* contains a set of *condition*s, *action*s and *decision rule*s (e.g. Table 1). Each condition consists of a *condition stub* (e.g. "People move Ear") and a *condition entry* (e.g. "Yes"). An example of condition is $\langle People\ move\ Ear, Yes\rangle$. Each action has an *action stub* (e.g. "Screen shows Message") and an *action entry* (e.g. "Message1" or ""). An example of action is $\langle Screen\ shows\ Message, Message1\rangle$. A decision rule is graphically represented as a table column. It is a combination of a set of conditions and actions. For instance, column 1 from Table 1 is a decision rule.

Table 1. A decision table example used in a ubiquitous system (example taken from [15])

Condition	1	2	3	4
People move Ear	Yes	No	Yes	No
Pressure on Crib	Yes	Yes	No	No
Action				
Screen shows Message	Message1			
iPhone rings			RingTone1	

A semantic decision table (SDT, [12]) is a decision table properly annotated with domain ontologies. According the Developing Ontology-Grounded Methods and Applications framework (DOGMA, [3] [10]), it contains three parts – a decision table, a set of annotated *lexons* and *commitments*.

A lexon is a binary fact type, which has the format of $\langle \gamma, t_1, r_1, r_2, t_2\rangle$. t_1 and t_2 are the two terms that represent two concepts in a natural language; r_1 and r_2 are the two roles that the concepts presented by t_1 and t_2 can possibly play with. γ is the context identifier that points to the document where $t_1$and t_2 are originally defined, and where r_1 and r_2 are meaningful. It can be, e.g. a URI. A lexon example is shown as follows.

$$\langle \gamma_1, Bunny, has, is\ of, Ear\rangle$$

A commitment specifies how to use a lexon or a set of lexons based on the agreements made by a user community. It can be an instantiation of a concept, a constraint on a lexon, grouping of lexons, an articulation that defines the mapping between terms and glosses, interpretation of role pairs, mapping of constraints and database schemas, and alignment of concepts.

The constraints in a commitment can be modeled in the Object Role Modeling language (ORM/ORM2 [1]). Initially, ORM is designed for modeling and querying databases at the conceptual level, enabling modeling, validating and maintaining database processes. The authors in [3] [10] discover that it can be used as a semantically rich modeling language for ontology because of its *visualization feasibility*, *expressive capability* in its graphical notation and *verbalization possibilities*.

We use Semantic Decision Rule Language (SDRule-L, [13]) to model SDT commitments. SDRule-L is an extension to ORM and hence follows similar modeling principles. Especially, it ensures the verbalization feasibility for any SDRule-L diagrams.

With regard to the verbalization issue, ORM has its root in Natural-language Information Analysis Method (NIAM, [6]), which provides a method of transforming verbalized information in a natural language into desired database models. SDRule-L adapts the idea of using the verbalization techniques to assist modeling ontological commitments for decision support. Each constraint corresponds to one sentence in a pseudo natural language.

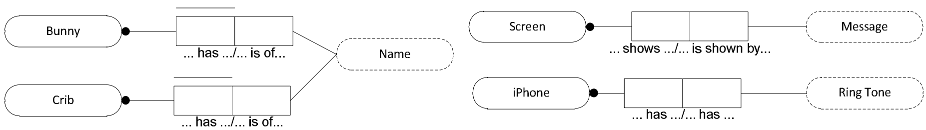


Fig. 1. SDT commitments for Table 1

For instance, Fig. 1 shows the SDT commitments for Table 1. Its verbalization is illustrated as follows.

EACH Bunny has EXACTLY ONE Name.
EACH Crib has EXACTLY ONE Name.
EACH Screen shows AT LEAST ONE Message
EACH iPhone has AT LEAST ONE Ring Tone.

In the following section, we will discuss our related work.

3 Related Work

V&V for decision tables has been attracting research interests since the birth of decision tables in computer science. The basic characteristics of decision tables are *completeness* and *correctness* [2]. The completeness of a decision table is ensured by exhaustively list all the possible combinations of conditions. Hence the completeness is a problem that can be easily solved. Yet the issue of correctness has always been a challenge. V&V for decision tables is to ensure the correctness of decision tables.

In early time, Shwayder [9] proposes using Quine-McCluskey method [5] to reduce redundancies in a table by combining equivalent decision columns. Pooch [7] provides a survey on algorithms for checking redundancy, contradiction and completeness within a table.

Recently, Vanthienen et al. [17] discuss the intra-tabular and inter-tabular anomalies. From their point of view, the intra-tabular anomaly is caused by a cyclic dependence between a condition and an action. The inter-tabular anomaly is caused by *redundancy*, *ambivalence* and *deficiency*.

With regard to V&V in general, we may as well use some classical V&V techniques for validating and verifying a (semantic) decision table. For instance,

- Turing test [16] uses textual input and output between a decision table based software and a rule auditor. The auditor considers a decision table valid if all the testing conditions and actions match his/her expectations.

- Face validity [8] is to ask a rule auditor whether the decision rules modeled by a decision modeler (or a group of decision modelers) reasonable or not.
- Quantitative validation [4] uses statistical models to evaluate the outputs of a decision table against either a set of randomly generated test cases or the test cases provided by a rule auditor.

How we proceed SDT V&V is as follows. We first transform a decision table into an SDT using the method illustrated in [11]. Then, we use the generated SDT commitments to validate and verify the SDT. If it satisfies all the commitments, then we say this SDT is valid and its original decision table is as well valid. If not, then both the SDT and its original table are invalid. In the following section, we will discuss how to check the validity of an SDT using ORM/ORM2 constraints.

4 ORM/ORM2 Constraints for SDT V&V

In this section, we will discuss ORM/ORM2 constraints in SDRule-L for SDT V&V.

Value Constraint

A value constraint is used to restrict allowed values for a condition entry. It can be *enumeration* (e.g. {a, b, c}), *range* (e.g. [0..100]) or *multiple* (e.g. {a, b, [0..100]}).

Table 2. A decision table on deciding whether or not to accept to process based on the value received from a temperature sensor, the age and the login state of a user

Condition	1	2	3	4	…	n	…
Age	>=18	>=18	>=18	>=18	…	**>=100, <=350**	
Temperature Sensor	>=0,<30	>=0,<30	>=0,<30	**>=-10,<0**	**…**	>=0,<30	
Login State	Yes	No	**Maybe**	Yes	…	Yes	…
Action							
Accept	*		*	*		*	

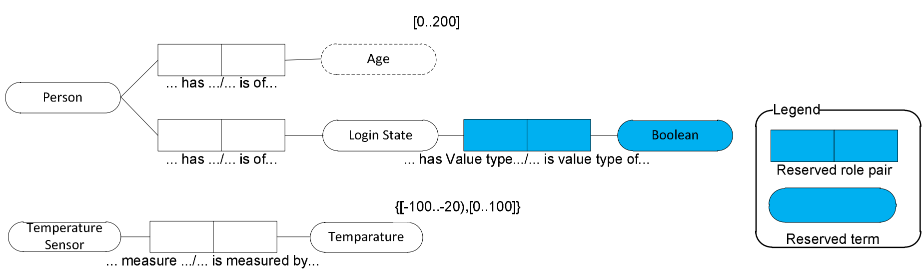


Fig. 2. SDT commitments for Table 2

Our SDRule-L engine does not only interpret value constraints as ORM/ORM does, but also accepts default value constraints defined by Float, Boolean and Integer. The value ranges of a Float and an Integer are $[Float.MIN, Float.MAX]$ and $[Integer.MIN, Integer.MAX]$. We use any of the following value enumerations as the default value constraints for a Boolean: {Yes, No}, {Y, N}, {True, False}, {T, F}, and {1, 0}. We may as well override the value ranges of these value types.

Table 2 is an example of a decision table, the SDT of which contains the designed commitments as illustrated in Fig. 2.The verbalization for Fig. 2 is illustrated as follows.

VALUE of Age is [0..200]
VALUE of Temperature Sensor is {[-100..-20), [0..100]}

When we use Fig. 2 to validate and verify Table 2, column 3 is considered invalid because "Login State" has Boolean values and "Maybe" is not a default Boolean value. The range of "Temperature Sensor" in column 4 is not in {[-100..-20), [0..100]} and hence is invalid. Column n contains "Age" that is beyond [0..200]; therefore, it is also invalid.

Occurrence Frequency

A frequency constraint is applied when we want to restrict the number of the instances of a role.

We use it to check the combination of conditions in a decision column. If certain conditions appear more than the allowed frequency, then the actions of this decision column must be empty.

Table 3. A decision table on deciding whether or not to turn on Actuator x based on the availability of X-Box557, X-Box120 and MS Xbox 360

Condition	1	2	3	4	5	6	7	8
X-Box 557	Yes	Yes	Yes	Yes	No	No	No	No
X-Box 120	Yes	Yes	No	No	Yes	Yes	No	No
MS Xbox 360	Yes	No	Yes	No	Yes	No	Yes	No
Action								
Actuator x	*		*		*	*		*

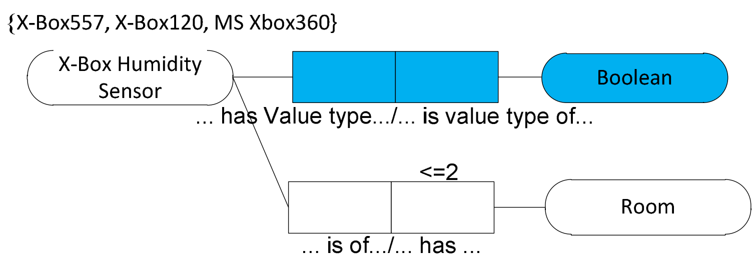


Fig. 3. SDT commitments for Table 3

Table 3 shows a decision table example, the SDT of which contains the commitments as shown in Fig. 3. The verbalization of Fig. 3 is shown as follows.

VALUE of X-Box Humidity Sensor is {X-Box557, X-Box120, MS Xbox360}.
EACH Room has AT MOST 2 X-Box Humidity Sensors.

Accordingly, column 1 from Table 3 does not satisfy the constraint and hence is invalid.

Mandatory

A lexon role can be *mandatory* or *optional*. A role is mandatory iff it is played by every member of the population of its connected object type, otherwise, it is optional ([1], p. 162). A mandatory constraint is equivalent to a "<=1" frequency constraint.

Mandatory constraint can be used to check whether a column that contains actions also contains *at least one* requested condition.

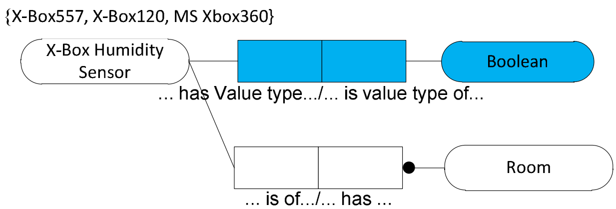


Fig. 4. SDT commitments for Table 3

For example, if we assign the commitments as illustrated in Fig. 4 to the SDT from Table 3, then column 8 in Table 3 is invalid. It is because this column contains an action whilst no "X-Box Humidity Sensor" is specified.

The verbalization for Fig. 4 is illustrated as below.

VALUE of X-Box Humidity Sensor is {X-Box557, X-Box120, MS Xbox360}.
EACH Room has AT LEAST ONE X-Box Humidity Sensor.

Table 4. A decision table on deciding whether or not to turn on Actuator x based on the availability of X-Box Humidity Sensors

Condition	**1**	**2**	**3**
X-Box Humidity Sensor	{X-Box557, X-Box120}	{X-Box557, MS Xbox360}	**N/A**
Action			
Actuator x	*		*

Table 4 is another example, on which we apply a mandatory constraint. It contains an object type as a condition stub and the value members of this object type as its condition entries. If we annotate the condition entry "N/A" with a notation of empty set, then column 3 in Table 4 is invalid because each room must have one X-Box Humidity Sensor.

Uniqueness

A uniqueness constraint is equivalent to a ">=1" frequency constraint.

For example, if we have the commitments as shown in Fig. 5 in the SDT from Table 3, then columns 1, 3 and 5 in Table 3 are invalid because there are actions in those columns and it contains more than one "X-Box Humidity Sensor".

We can verbalize the SDT commitments in Fig. 5 as follows.

VALUE of X-Box Humidity Sensor is {X-Box557, X-Box120, MS Xbox360}.
EACH Room has AT MOST ONE X-Box Humidity Sensor.

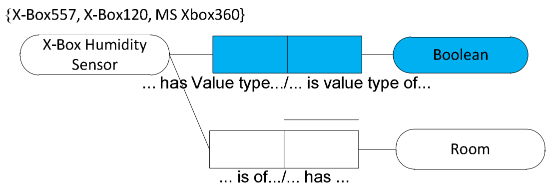


Fig. 5. SDT commitment for Table 3

Table 5. A decision table on deciding whether or not to turn on Actuator x and Actuator y based on the availability of Sensors

Condition	1	2	3	…	n
Humidity Sensor	{X-Box557, X-Box120}	{ MS Xbox360}	{X-Box557, Xbox360}	…	{ MS Xbox360}
Sensor	{EZEYE 1011A}	{EZEYE 1011A}	{**X-Box120**}	…	{E1, **X-Box557**}
Action					
Actuator x	*		*		
Actuator y		*	*		*

The uniqueness constraint can be used not only with Boolean condition entries but also with set condition entries. Table 5 shows an example of condition entries as sets. Fig. 6 is the commitments in its SDT. We can see that X-Box120 and X-Box557 are defined as the members for both Humidity Sensor and Sensor, and, only one Humidity Sensor is allowed in one room. Therefore, columns 3 and n in Table 5 are invalid because each of them contains (an) action(s) and two Humidity Sensors.

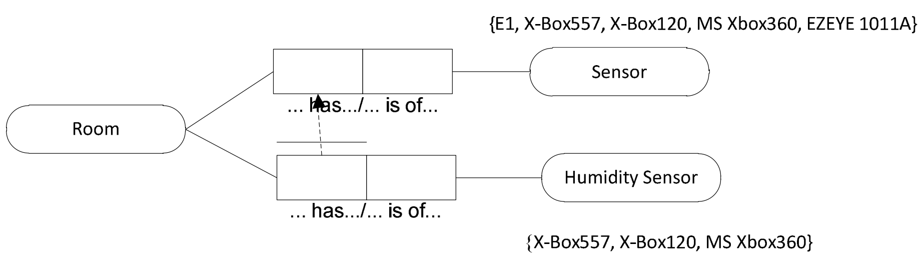


Fig. 6. SDT commitments for Table 5

The verbalization for the commitments in Fig. 6 is as follows.

VALUE of Humidity Sensor is {X-Box557, X-Box120, MS Xbox360}
EACH Room has AT MOST ONE Humidity Sensor
VALUE of Sensor is {E1, X-Box557, X-Box120, MS Xbox360, EZEYE 1011A}

Exclusive-Or and Subtyping

An exclusive-or constraint is applied when we want to ensure two sets do not overlap each other. We use it for checking the exclusivity between two conditions or two actions. Note that the condition entries of these conditions must be Boolean values. If we want to check the exclusivity between two actions, then we need to combine a constraint of exclusive-or and subtyping.

Table 6 shows a decision table example, the SDT of which contains the commitments as illustrated in Fig. 7. Its verbalization is illustrated as follows.

EACH Room has EITHER Humidity Sensor OR Light Sensor, BUT NOT BOTH
EACH ACTION can EITHER BE X OR Y, BUT NOT BOTH
X IS SUBTYPE OF ACTION
VALUE of X is {Actuator X}
Y IS SUBTYPE OF ACTION
VALUE of Y is {Actuator Y}

Table 6. A decision table on deciding whether or not to turn on Actuator x and Actuator y based on the availability of Humidity Sensor and Light Sensor

Condition	1	2	3	4
Humidity Sensor	Yes	Yes	No	No
Light Sensor	Yes	No	Yes	No
Action				
Actuator x	*		*	
Actuator y			*	

Column 1 in Table 6 is invalid because it does not satisfy the constraint "EACH Room has EITHER Humidity Sensor OR Light Sensor, BUT NOT BOTH" and it contains an action. Column 3 in Table 6 is invalid because it does not satisfy the constraint "EACH ACTION can EITHER BE Actuator x OR Actuator y, BUT NOT BOTH".

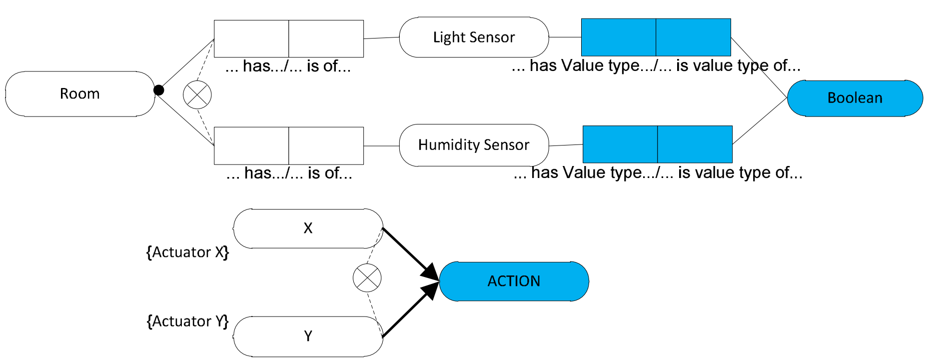


Fig. 7. SDT commitments for Table 6

We can also use subtyping to check the dependency between two conditions when their condition entries are Boolean values. Suppose we have a decision table as shown in Table 7, the SDT of which contains the commitments as illustrated in Fig. 8. Its verbalization is shown as follows.

Humidity Sensor IS SUBTYPE OF Sensor

Column 2 in Table 7 contains the conditions $\langle Humidity\ Sensor, Yes\rangle$ and $\langle Sensor, No\rangle$, and an action. According to Fig. 8, Humidity Sensor is a subtype of Sensor, which means that the existence of Humidity Sensor implies the existence of Sensor. Therefore, this column is invalid.

Table 7. A decision table on deciding whether or not to turn on Actuator x based on the availability of Humidity Sensor and Sensor

Condition	1	2	3	4
Humidity Sensor	Yes	Yes	No	No
Sensor	Yes	No	Yes	No
Action				
Actuator x	*	*	*	

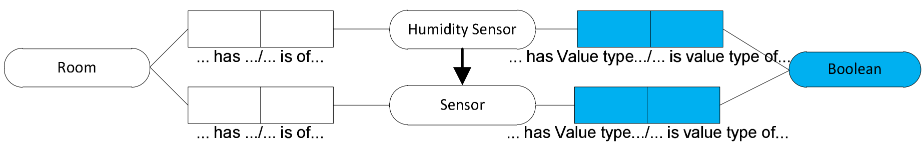


Fig. 8. SDT commitments for Table 7

5 Conclusion and Future Work

In this paper, we have discussed how SDRule-L directly uses ORM/ORM2 constraints to validate and verify SDTs/decision tables. These constraints are value, uniqueness, mandatory, frequency, exclusive-or and subtyping.

We have designed an SDT rule engine for V&V using these constraints. Note that the V&V methodology, which is out of the paper scope, is one of our future works.

SDRule-L is a language to model SDT commitments, which are the input of the SDT rule engine. ORM/ORM2 is a graphical language whilst SDRule-L is both a graphical language and a textual language, which has a given syntax and semantics. SDRule-L is designed for being used by both business rule modelers (people) and the SDT rule engine (machine). Like ORM/ORM2, each SDRule-L commitment can be verbalized in order to assist the business rule modelers.

We are still working on the implementation of the SDT rule engine. We use a markup language called SDRule-ML to publish and share SDT commitments designed in SDRule-L. An initial design on how to store the discussed SDT commitments in SDRule-ML and its mapping to RDFs/OWL is recorded in [14]. We are currently busy with the implementation of the mapping as well.

In this paper, we focus on the V&V issues within one decision table. In the future, we will study the V&V issues across decision tables.

Acknowledgments. The work has been supported by the EU ITEA-2 Project 2008005 "Do-it-Yourself Smart Experiences", founded by IWT 459.

References

[1] Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn., March 17. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann (2008) ISBN-10: 0123735688, ISBN-13: 978-0123735683

[2] Henry Beitz, E., Buck, N.H., Jorgensen, P.C., Larson, L., Maes, R., Marselos, N.L., Muntz, C., Rabin, J., Reinwald, L.T., Verhelst, M.: A modern appraisal of decision tables, a Codasyl report. ACM, New York (1982)

[3] Jarrar, M., Meersman, R.: Formal Ontology Engineering in the DOGMA Approach. In: OTM 2002, pp. 1238–1254 (2002)

[4] Lehner, P.E.: Towards an empirical approach to evaluating the knowledge base of an expert system. IEEE Transactions on Systems, Man and Cybernetics 19(4), 643–655 (1989)

[5] McCluskey, E.J., Schorr, H.: Minimization of Boolean functions. Bell Syst. Tech. J. 35(5), 1417–1444 (1956)

[6] Nijssen, G.M., Halpin, T.A.: Conceptual schema and relational database design: a fact oriented approach. Prentice Hall (1989) ISBN: 0-13-167263-0

[7] Pooch, U.W.: Translation of Decision Tables. Journal of ACM Computing Surveys (CSUR) Surveys 6(2) (June 1974)

[8] Sargent, R.G.: Verification and validation of simulation models. In: Proc. of the 37th Conference on Winter Simulation, WSC 2005 (2005) ISBN:0-7803-9519-0

[9] Shwayder, K.: Combining decision rules in a decision table. Communications of the ACM 18(8) (August 1975)

[10] Spyns, P., Meersman, R., Jarrar, M.: Data modeling versus Ontology engineering. SIGMOD Record: Special Issue on Semantic Web and Data Management 31(4), 12–17 (2002)

[11] Tang, Y.: On Conducting a Decision Group to Construct Semantic Decision Tables. In: Chung, S., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 534–543. Springer, Heidelberg (2007)

[12] Tang, Y., Meersman, R.: Towards Building Semantic Decision Tables with Domain Ontologies. In: Chan, M.C., et al. (eds.) Challenges in Information Technology Management. World Scientific (2008) ISBN 978-981-281-906-2, 981-281-906-1

[13] Tang, Y., Meersman, R.: SDRule Markup Language: Towards Modeling and Interchanging Ontological Commitments for Semantic Decision Making. In: Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches. IGI Publishing, USA (2009) ISBN: 1-60566-402-2

[14] Tang, Y., Meersman, R.: Towards Directly Applied Ontological Constraints in a Semantic Decision Table. Submitted to RuleML 2011 (2011)

[15] Tang, Y., Debruyne, C., Criel, J.: Onto-DIY: A Flexible and Idea Inspiring Ontology-Based Do-It-Yourself Architecture for Managing Data Semantics and Semantic Data. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6427, pp. 1036–1043. Springer, Heidelberg (2010)

[16] Turing, A.: Computing Machinery and Intelligence. Mind LIX (236), 433–460 (1950)

[17] Vanthienen, J., Mues, C., Aerts, A.: An Illustration of Verification and Validation in the Modeling Phase of KBS development. Journal of Data & Knowledge Engineering 27(3), 337–352 (1998)

Roles in ORM: A Suggested Semantics

Andy Carver

INTI International University, Malaysia
andy.carver@newinti.edu.my

Abstract. Evidence grows that ORM has some problems to overcome in order to escape the charge that it cannot reliably lead, as claimed, to ONF (optimal normal form) relational schemas. Some of that evidence we present here. We also indicate that there is more than one way to address the issue effectively, thus saving ORM's reputation as a generator of fully normalized relational schemas. But of two ways mentioned, the more satisfactory one involves, essentially, ascribing a certain semantics to the idea of "role-playing" in ORM. We show how it would address these issues, and why it is the better approach.

1 Introduction

As an information-system design method, ORM has long been touted as able to produce, reliably and by application of an algorithm, Fifth Normal Form (5NF) relational database schemas, free of redundancy. (Indeed, it produces Optimal Normal Form (ONF) schemas, which implies also a minimum number of mapped relational table schemes.) This is certainly an important claim for ORM's marketing message. However, increasingly it appears doubtful that this claim is entirely accurate. The need for "pre-Rmapping" transformations, or for "fixes" to Rmap, in order to rid the mapped relational schema of redundancy, has increasingly loomed. Indeed, as we will show early in this paper, although the problems in this regard have become familiar somewhat, they are broader than has been made clear from the well-known examples.

That is not to suggest that such problems are hard to "fix". But the usual approach to "fixing" these problems, which is to add some transforms or manipulations to the Rmap algorithm so as to produce a clean, 5NF schema, leave open the question of whether ORM is faultless in the matter: Do not these "fixes", it may be asked, simply patch the conceptual ORM schema? If so, what becomes of that claim for ORM?

Moreover, there is another approach to addressing these problems that we will argue is, for this and other reasons, more satisfactory. Nor does it do away with any convenient ORM constructs or transformations. That approach is to give "role-playing" in ORM a particular, definite meaning which we shall specify in this paper.

The plan of the paper is as follows. In section 2 we indicate the seeming lack of, but need for, specifics on what we mean in ORM by designating a particular object a "role-player" in a given fact. In section 3 we point out that the well-known difficulty of Rmapping nested entity types lacking fully-spanning uniqueness constraints (UCs) is just a special case of a more general problem that transcends nesting altogether and which demands that which we call "reference-scheme reduction" transforms, pre-Rmap—or equivalent "fixes" to Rmap. It would be incorrect, therefore, to suppose we

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 360–369, 2011.
© Springer-Verlag Berlin Heidelberg 2011

need only to "fix" Rmap so as to support certain nested ORM constructs, or that the only fixes ever needed will be for such new (or newly-permitted) ORM constructs. In section 4, we specify and offer our own definite semantics for "role-playing" in ORM. In section 5, we explain how this semantics addresses all the aforementioned perceived issues with ORM. In section 6, we argue that the approach which this semantics affords for addressing these issues, has advantages that make it more satisfactory, for addressing them, than is the offering of pre-Rmap transformations or the claiming to devise "fixes" for ORM or for Rmap. In section 7 we summarize our conclusions.

2 The Need for a Definite Meaning of "Role-Playing" in ORM

One of the most accepted ideas about ORM is that "facts of interest" include only those in which an object or objects play one or more roles. However, though this idea of "role-playing" by objects is taken as axiomatic, it is never suggested that all mentioned objects must or should play roles. It is quite normal for some objects not to be (taken as) role-players.

But there is little if any discussion in the ORM literature, as to the reason for, or the meaning of, the attribution of a fact "role" to an object. Is it arbitrary, which objects are taken as role-players? (ORM's allowing *compidot*s, i.e. *comp*ositely *id*entified *o*bject *t*ypes, increases the difficulty of this question.) In other words, what, if anything, is the difference between an object taken as playing a role in the fact (or in parallel with this, an object-mention as filling an object-hole in the sentence) and an object mentioned in the same fact's statement but not taken as playing a "role"?

Is there a specific meaning to be ascribed to it, when we attribute a "role" to an object? Or is the "role" merely an operationally convenient notation? That is, does its use or non-use, for a particular object in a particular part of a given fact, have nothing at all to do with the meaning incorporated in that fact, and find its basis rather in some unrelated aspect of the system design—for example, in an implementation decision? If the latter, it is hard to think of ORM as being a truly "conceptual" modeling method; for it works, at least partly, at some non-conceptual level, in every fact type of interest. To maintain and support ORM's integrity as "conceptual" method, then, it would seem necessary that there be some actual shade or difference of meaning behind, and signified by, the attribution of roles to objects, to distinguish their function or status from that of objects mentioned in the fact's statement but not attributed roles in it.

If we take this as an at least valid inference, let us explore what meaning it might make sense to us to ascribe to such attributions, given ORM's basis in predicate logic and given one fundamental principle from information theory. But to motivate that, we wish to fill out first the picture of what data redundancy issues can occur in ORM.

3 The Need for ORM Reference-Scheme Reduction Transforms

It has been shown that an essentially "ORM-like" approach to information system design is, perhaps surprisingly, a necessary condition for being sure of getting or having a 5NF relational schema [2]. However, that this approach is also a *sufficient* condition for such an outcome, is no longer apparently true: indeed, as ORM seems to stand now, it is false, as we shall see. And other data-redundancy anomalies, which fall short of non-normalization, have already been shown to attend the relational mapping of some ORM schemas, and to call for pre-Rmap use of “reduction transforms” [4].

Another sort of pre-Rmap transform, the one for nested fact types having non-spanning UCs, has been around longer and is well known [3, pp. 6-7; 5, pp. 145, 523]. However, it is not so well known that this is a special case of a broader category of problematic constructs that does not in general entail nesting. It seems useful to discuss this broader category, both in its own right and in order to correct the misperception many may have, that “needed pre-Rmap transforms” or “Rmap fixes” to support these nested constructs are necessary only due to a new, less-restrictive grammar which must (of course) be comprehended by the mapping procedure. Looser syntax is in fact not the problem.

Rather, what’s causing redundancy in these situations is that *the composite reference-scheme is not entirely referential:* some of it is being used to convey other, non-referential information. In some cases this may cause part of the composite reference to be redundant. The problem may be fixed by a suitable transformation that will externalize from the reference scheme, the fact type(s) needed to store this non-referential information.

How, one may ask, would we know that there is some “non-referential” usage being made, of a set of fact types that clearly make up a single, composite reference scheme? That usage could be detected by existence of a valid external constraint, other than the “preferred-identifier” UC, spanning roles only within this reference scheme. The classic pattern of this, and the one that causes data redundancy problems, is shown in Fig. 1. The essential element of this pattern is not the exact constellation of the reference scheme’s fact types, but rather an added, non-preferred-identifier external UC (indicated here by the arrow).

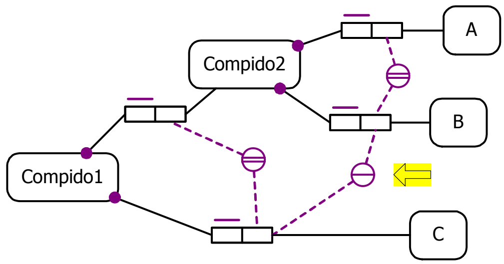


Fig. 1. The general pattern that calls for reference-scheme reduction

Note that the compidots in this pattern may or may not be nested entity types. This does not affect the pattern, nor the resulting data redundancy. However, only with nested entity types can we have the peculiar situation that our composite reference scheme has an *internal* non-preferred-identifier UC causing the same issue. The two transforms shown in Fig. 2 illustrate the parallels between the pattern (and its transform) for the external UC, and the same for the internal, non-spanning UC.

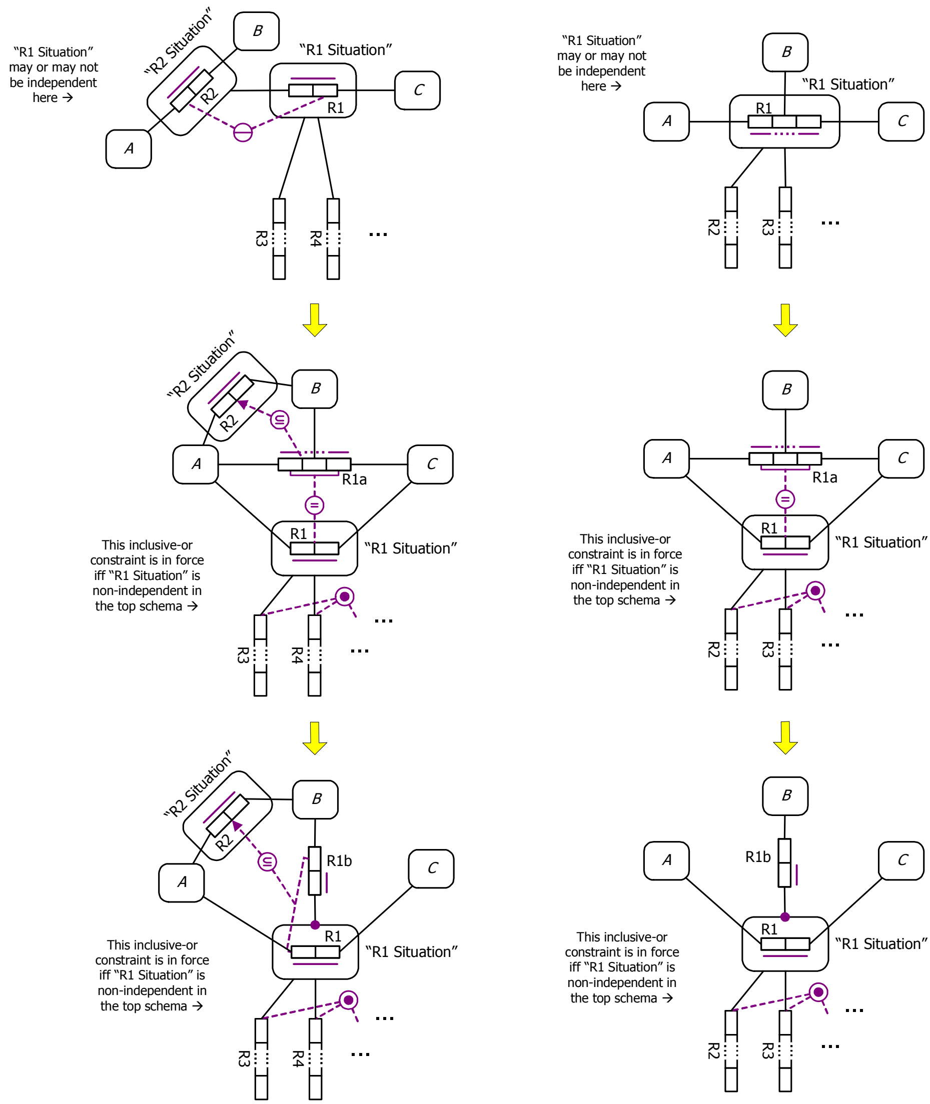


Fig. 2. Two reference-scheme reduction transforms involving nesting

Note that if the nested fact type on the right-hand side of Fig. 2 is binary rather than n-ary, we have the familiar special case, which involves a further transform to an even simpler schema. This is shown in Fig. 3.

The seriousness of the problem these data redundancies can cause is illustrated by the example in Fig. 4. The extra UC is in Semester Enrolment's reference scheme.

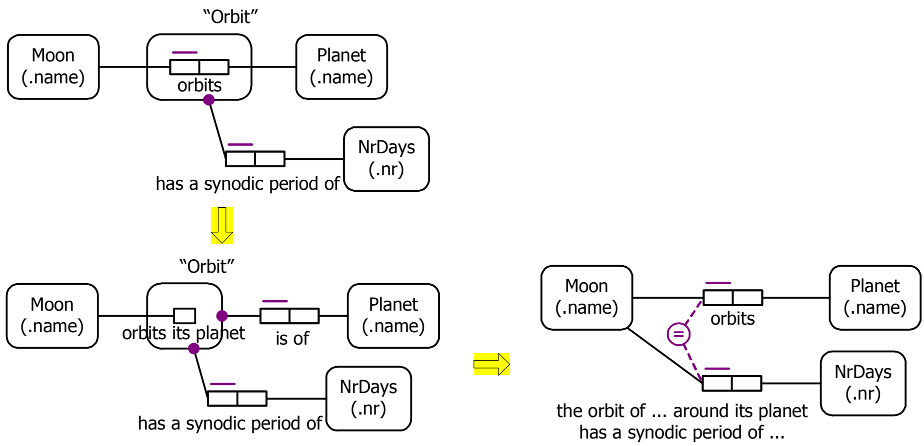


Fig. 3. Special case of such transforms, involving nesting of a binary

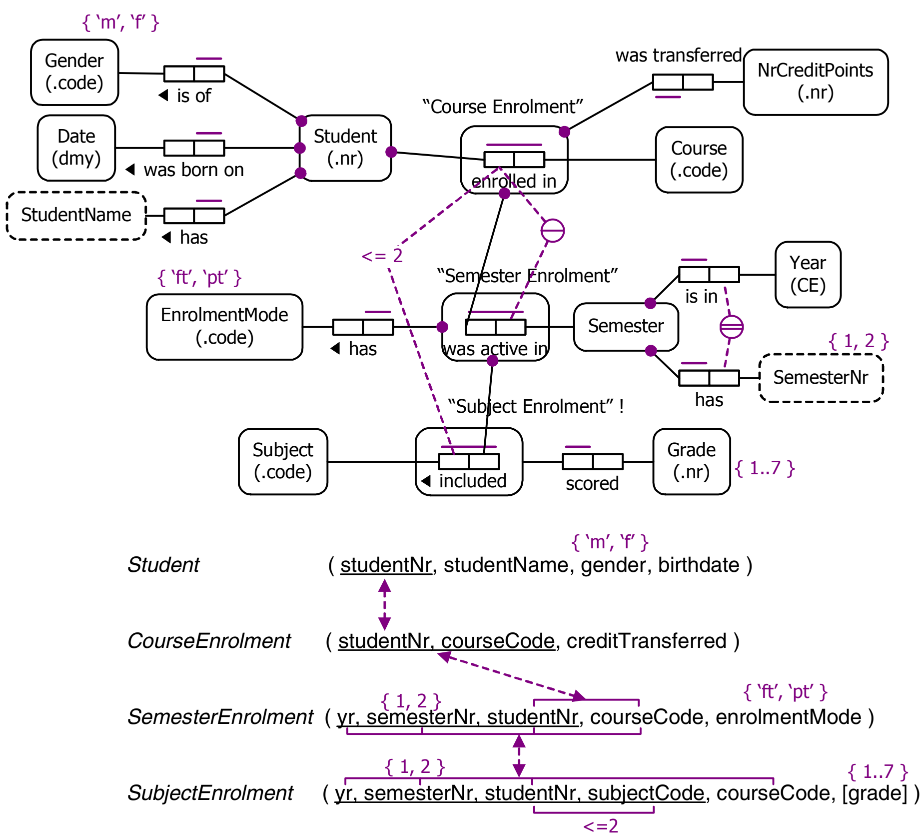


Fig. 4. Example of ORM schema that Rmaps to a non-normalized relational schema

This ORM conceptual schema Rmaps to the non-normalized relational schema shown below it. The "Subject Enrolment" table is N2NF (i.e. not in Second Normal Form) as a result of the external UC. So, the problem in this case did *not* arise from a non-spanning, internal UC on a nested fact type. The latter is only a special case of this. So again, we must seek a solution(s) that goes beyond mere support for newly-permitted constructs in ORM (such as nesting that involves non-spanning UCs). The problem here is with ORM, as so defined; or with Rmap, as generally defined.

As we have noted, the usual response to such problems is to say that "Rmap needs fixing"; and to suggest fixes. And again, we are not suggesting that such "fixes" will not really fix. But if the fixes work, they turn out to be equivalent to performing some variant of the above reduction transform. We do suggest, moreover, that they leave ORM open to the criticism that it itself, not Rmap, is to blame. Such criticism may be defused, we claim, by specifying the meaning of attributing "role-playing". This will also implicitly rationalize the use of reduction transforms, as a *normal* part of ORM.

4 Our Suggested Semantics for Attribution of "Role-Playing"

Here is our suggested meaning of attribution of "role-playing" to an object, in ORM:

> A specific object is said to be "playing a *role"* in a given fact, in ORM, where and only where the mention of that object in that fact's statement is, in its context, presumably intended as a *maximal completely* information-bearing object-reference. Or, in terms from speech-act theory, we attribute a role to an object exactly where the latter intent is the presumable *perlocutionary* sense (or "force") of the object's mention.

A basic principle of information theory describes the inverse relationship between informativeness and predictability: if the information-item that will appear in a particular place in an expression is entirely predictable, it carries no information. Likewise in a database, data that is entirely predictable given its context (i.e. the data, schema, and "domain of discourse" [7, p. 672; 6, p. 80n.]) is redundant. The notion of predictability here parallels substantially the "bound" character of (some) individual-variable occurrences in predicate logic. The meanings of the terms "maximal" and "completely" here are as defined in [1]: these qualifiers are needed due to ORM's allowance of compidots (compositely-identified-object types), and require informativeness of the composite reference *entire,* which is what we mean by "completely"; whereas "maximal" means that the object reference is not contained within any other completely information-bearing object reference. All this is further discussed in [1].

5 How Using This Semantics Addresses Such Redundancy Issues

The idea of ORM *reduction transforms* was introduced in a 2007 ORM Workshop paper by Dr. Terry Halpin, Mr. Kevin Owen, and this author [4]. The paper suggested that such transforms should always be performed, on any conceptual schema intended for Rmapping, prior to the Rmapping. The idea of ORM *pre-Rmap schema transforms,* obligatory to invoke before undertaking any Rmapping, had been appealed to earlier in the ORM literature, but only occasionally and ad hoc.

Nor did our 2007 paper address the underlying problem(s) necessitating reduction transforms. The underlying cause for the sort of data-redundancy addressed by these reduction transforms was explored by this author's 2008 ORM Workshop paper [1]. There we were able to elucidate that the "reduction transforms" described in our 2007 ORM Workshop paper all addressed cases of "redundant object-reference". Also we were able to specify a sub-procedure, to be deployed during the early part of the ORM Conceptual Schema Design Procedure (CSDP), that enables only-non-redundancy-causing attribution of "roles" to objects (where the latter are mentioned in the sample facts); it does the latter by determining which object-references are *maximal completely information-bearing* ones, and attributing a "role" to the object only in, and at, such referential cases.

We then suggested that by using this procedure, we obviate the need for "reduction transforms" by eliminating every occasion for their use. But it may not be immediately clear, even to one who has read the 2008 paper, how that procedure would obviate the need for the new, "reference-scheme reduction" transform we have shown above. Perhaps we may clarify this by way of an illustration. Consider the example in which "Orbit" is an entity type that objectifies "Moon orbits Planet" and plays a role in "Orbit lasts for NrOfEarthDays". Suppose we began our modeling, in Step 1, working with the sample fact-statement "the moon 'Luna' orbits the planet 'Earth', which orbit lasts for the nr-of-earth-days 30", and possibly other sample fact-statements.

Now, the "..., which orbit..." anaphoric reference in the above statement (which reference expresses the objectification being done by the nested entity type) needs, in order to be stored in the database, to be replaced by a non-anaphoric reference, i.e. by an *explicit* reference to the "orbit" being mentioned. Our CSDP Step 1 procedure enforces this by requiring all maximal, completely information-bearing references—*even anaphoric ones*—to be unpacked into definite descriptions.

But then, *how* should we verbalize that orbit reference explicitly as a definite description? We can unpack that second part of the sentence into either (but not both) of the following two meaning-possibilities (in which we have left non-proper nouns uncapitalized in order not to beg the issue, to which objects are roles rightly attributed):

1) "the orbit *of the moon named 'Luna' around the planet named 'Earth'* lasts for the nr-of-earth-days 30"; or,
2) "the orbit *of the moon named 'Luna' around its planet* lasts for the nr-of-earth-days 30".

There is no third, alternative referential sense that holds any plausibility.

Now, the first of these two verbalizations assumes that the fact that Luna orbits Earth is of interest (otherwise, why mention it?). But either the latter fact is actually stored (or to be stored), elsewhere in the context, or it is not. If it is stored elsewhere, then the UC that each moon orbits at most one planet relegates any "planet" reference within the "nr-of-earth-days" fact to being a redundant, secondary, non-info-bearing (because completely predictable) object reference. (So the role would be considered as played by Moon.)

And if it is *not* stored elsewhere, then since we have a fact of interest that is mentioned inside this definite description of the orbit, we find that the purpose of this definite description is conflated: it is "neither fish nor fowl", for it is neither a purely referential, *object* term, nor is it clearly intended as a conveying of that "planet" information about the orbit. In other words, the verbalization leaves ambiguous the intended purpose (i.e., the "perlocutionary sense") of this part of the statement; indeed, it seems as if the speaker may be ambivalent and unclear about what he or she is trying to accomplish by this part of the statement. To be unambiguous in purpose, it should be rephrased: e.g., "the orbit *of the moon named 'Luna', which is around the planet named 'Earth',* lasts for the nr-of-earth-days 30". Clearly, here we would be stating two different things about the orbit. Also clearly, we would be identifying the orbit by the moon doing it—and not at all by the planet being orbited. So the latter information would become external to the definite description. This lands us back in the case of having that fact stored separately. Thus, the composite reference, in our first verbalization, to this particular "orbit" turns out to be *not completely* information-bearing (since the reference it contains to its planet, "Earth", is not), thus requiring, as it were, role-redirection (to Moon again) to satisfy our CSDP Step 1 procedure.

On the other hand, the second of the above verbalizations assumes that either the fact that Luna orbits Earth is stored elsewhere, *or that this fact is not of interest.* So "its planet" (which actually means "the planet of it") is taken as an anaphoric, non-information-bearing object term. In either verbalization therefore, the overall reference to the orbit object is non-completely informative.

But that non-informativeness seems to be built into *every* reference to the orbit, since it is built into that object's very reference scheme. (Even if we want just to list orbits, independently of their playing a role in a non-reference-scheme fact type, we will still have to deal with the issue of what is required simply to reference an orbit without verbalizing any non-referential information. So, we will need to omit from this set of "existential" facts any reference to the "planet", lest the fact type conflate, again, existential and non-existential information.) So, by our new criterion of "role-playing", it appears the object can play no role! But of course, we do want object types to be capable of role-playing. Thus, the Orbit object type's composite reference scheme is itself invalidated, in effect, by the criterion of "role-playing" we have adopted.

6 Why This Semantics Is the Best Way to Address Such Issues

We ORM advocates would like to be able to say that, when used correctly and Rmapped, ORM is a methodology sufficient to guarantee an ONF relational database schema. But seemingly ORM needs "fixing", somehow, in order to support this. However, an approach to "fixing" it that involves patching some additional manipulations onto the Rmap procedure, is one by which ORM cannot escape all taint of inadequacy. Let us explain why.

The only such Rmap fixes that would be adequate to the problem would be, we have implicitly shown, ones that re-arrange the fact types and reduce the reference schemes. Now, this must not be done later than a point prior to any grouping of fact types into relational tables. Since the latter takes place in Steps 1 and 2, these "fixes" to Rmap cannot be applied, in general, later than Step 0. Already in [5, p. 510], the special transform for nested fact types lacking fully-spanning UCs is placed at Step 0.7. But the reference scheme reduction transforms must, in general, take place even before Step 0.2, because they alter composite reference schemes (which must be "mentally erased" within that Step). Indeed, any relevant reduction transforms must happen before they have any effect on other parts of the procedure. Thus, for all practical purposes, they might as well happen at the very beginning of the procedure.

But this means that to call them "fixes to Rmap", or even part of Rmap, is entirely arbitrary. They could more simply, and just as well, be considered and used as "pre-Rmap" transforms (as some ORM practitioners do consider them). This is exactly why ORM itself cannot evade criticism, based simply on the suggestion that Rmap "needs fixing". ORM itself could still, and reasonably, be thought to be the problem.

6.1 What It Would Take for ORM to Escape Any Taint from These Problems

In order to allow, as practically we must allow, some ORM constructs that cause data redundancy, and yet also to allow ORM itself to escape any taint of fault for such redundancies, it must make a distinction that some other formal languages make. We must distinguish, in a word, a "core" or "canonical" subset of ORM. Such a subset of a language is both stricter and simpler, in grammar, than is the language itself.

For an example of a language that makes such a distinction, consider XQuery. A subset of XQuery is distinguished as "Core XQuery" [8]. Among XML languages, XQuery's most distinctive expression type is the FLWOR expression; this is the XQuery analog to SQL's SELECT statement. "FLWOR", usually pronounced "flower", is an acronym for the types of clauses that may appear within such expressions: "For" clause, "Let" clause, "Where" clause, "Order by" clause, and "Return" clause. But the grammar for the expression is stricter in Core XQuery than in XQuery: In XQuery, each FLWOR expression may have as many For clauses and Let clauses as desired, in any order (at the start of the FLWOR); also, each such clause may contain any number of variable-bindings. But in Core XQuery, a FLWOR expression contains at most one Let, or one For clause, and not both; and that clause may contain at most one variable-binding. Each FLWOR expression, before processing, gets "normalized" to Core XQuery by a transformation performed by the XQuery processor.

An exactly parallel situation is what we suggest for ORM. Once the proper distinction is made between ORM and Core (or "Canonical") ORM, the transforms from ORM to Core ORM will have become not only normal but expected, even predicted.

6.2 What It Would Take to Distinguish Correctly a "Canonical" ORM Subset

The difference between a well-formed expression in a language, and a legal construct of the core, canonical language, is that the former has no semantics except as it is also in the core language or can be translated into an expression in the core language. That is to say, a well-formed expression that is however not part of the core language, has no informal, real-world semantics, but rather its only "meaning" is, the core-language expression into which it translates (or "normalizes"). It is a "defined symbol".

All ORM constructs, however, already are to be mapped to things in the real world. There is no way at this juncture, to divest any of them of their informal meaning. Nor would we wish to do so: at least, not directly.

6.3 What Would Suffice to Divest Anomalous ORM Structures of Meaning

But the same effect may be achieved, implicitly, by adopting the meaning we are suggesting for “role-playing”. For if we adopt this meaning, it is one that is implied every time we draw a role-box in ORM—just as, for example, all primitive entity types in an ORM schema are implicitly mutually exclusive. If so, then an ORM diagram that attributes a role to a kind of object where and for which, according to the context in the schema, the objects could not possibly be “role-players” by this definition, we have a self-contradictory construct. Thus, such a construct (i.e. any to which a reduction transform applies) is *ipso facto* divested of informal, i.e. real-world meaning. Its only meaning is that given by its reduction transform, viz. its “normalized” form.

7 Conclusion

If we wish to continue to claim that ORM leads reliably to ONF relational schemas, we need to take steps to support that. There is more than one way to do this. But the only way that makes the least structural or conceptual impact on ORM, or which leaves it invulnerable to any taint of inadequacy to supply ONF, is the approach that adopts, for “role-playing”, the semantics described herein. We urge its adoption.

References

1. Carver, A.: How To Avoid Redundant Object-References. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 770–779. Springer, Heidelberg (2008)
2. Carver, A., Halpin, T.: Atomicity and Semantic Normalization. International Journal of Information System Modeling and Design 1(2), 23–39 (2010)
3. Halpin, T.: What is an Elementary Fact? In: Nijssen, G.M., Sharp, J. (eds.) Proc. 1st NIAM-ISDM Conf. (1993)
4. Halpin, T., Carver, A., Owen, K.M.: Reduction Transformations in ORM. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 699–708. Springer, Heidelberg (2007)
5. Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2008)
6. Levinson, S.C.: Pragmatics. Cambridge University Press, Cambridge (1983)
7. Lyons, J.: Semantics, vol. 2. Cambridge University Press, Cambridge (1977)
8. The World Wide Web Consortium: XQuery 1.0 and XPath 2.0 Formal Semantics, `http://www.w3.org/TR/xquery-semantics/`

Formal Specification of a Meta Hierarchical Logical Data Model Using Object Role Modeling

Robert J.L. Schmaal[1], Herman Balsters[2], and Serge Valera[1]

[1] Software Systems Division, European Space Research and Technology Centre, Noordwijk, The Netherlands
{robert.schmaal,serge.valera}@esa.int
[2] Faculty of Economics and Business, University of Groningen, Groningen, The Netherlands
h.balsters@rug.nl

Abstract. In the past, numerous ways have been presented of mapping Object Role Modeling (ORM) conceptual schemas to relational logical schemas. However, mapping to hierarchical logical schemas has received little attention. This is unfortunate, since hierarchical logical schemas can be used to acquire physical schemas, like hierarchical DDL or XSD documents, along the lines of a three-level data modeling architecture. Given that hierarchical databases still exist abundantly in industrial practice, mapping ORM conceptual schemas to hierarchical logical schemas can be very relevant. A meta model description of a hierarchical logical data model offers a formal basis for defining such a mapping. We note that a hierarchical logical data model, as well as a meta model to which it should conform, is often described only informally in literature, particularly lacking a formal meta model specification. In this paper, we present a formal specification of a meta hierarchical logical data model using ORM.

Keywords: meta model, hierarchical logical data model, logical schema, conceptual schema, Object Role Modeling.

1 Introduction

Object Role Modeling (ORM) is a fact-oriented approach for modeling information in terms of facts, where facts and rules may be verbalized (and hence validated) in a language easily understandable by non-technical domain experts. In contrast to Entity-Relationship (ER) modeling [1] and Unified Modeling Language (UML) class diagrams [2], ORM models are attribute-free, treating all facts as relationships (unary, binary, ternary etc.). ORM does, however, include procedures (e.g. the RMap [3]) for mapping to attribute-based structures, such as those of ER or UML. For a comparison of ORM with UML see [4].

We will use the term fact-based modeling (FBM) as the general name of several fact-based conceptual data modeling dialects, such as ORM, Natural language Information Analysis Method (NIAM) [5], and Fully-Communication Oriented Information Modeling (FCO-IM) [6].

In the past, numerous ways have been presented of mapping FBM conceptual schemas to different types of relational logical schemas (e.g. relational models), in-

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 370–379, 2011.
© Springer-Verlag Berlin Heidelberg 2011

cluding the well-known R-map algorithm [3] of ORM. Mapping to hierarchical logical schemas, however, has received little attention. This is unfortunate, since hierarchical logical schemas can be used to acquire physical schemas, like hierarchical DDL or XSD documents, along the lines of a three-level data modeling architecture. According to such an architecture, data modeling proceeds from the conceptual level down to the logical level, to finally arrive at the physical level. At each level, respectively conceptual, logical and physical schemas are acquired [7]. A conceptual schema describes the Universe of Discourse in terms of human-oriented concepts. In turn, a logical schema groups information into structures supported by the generic logical architecture (e.g. network, hierarchical, relational). Finally, a physical schema specifies implementation specific physical storage and access structures [3].

We also mention that many hierarchical databases (e.g. Information Management System (IMS) [8]) still exist in industrial practice; indeed, if your set of database transactions is known beforehand, and is expected to remain fairly stable in the future, hierarchical databases might even be your best choice for database implementation.

Mapping ORM conceptual schemas to XSD documents has been described in [9]; the major difference between our work and [9] is that in [9] the authors refrain from using an explicit logical data model to eventually acquire an XSD document, hence not abiding to a three-level data modelling architecture. We note that we regard an XSD document as a physical schema, and not as a logical schema.

Other studies have proposed methods for improving the comprehensibility of ORM conceptual schemas [10,11,12], involving notions of hierarchies. However, none of these methods result in a mapping from an ORM conceptual schema to a hierarchical logical schema.

Hierarchical data modeling is also considered adequate for the European Space Agency (ESA); especially for developing interfaces to exchange spacecraft data and knowledge between industries and agencies, for developing Man Machine Interfaces, but also to structure ORM conceptual schemas for facilitating the management of large data models. ESA is also involved in the development of FBM tools for global conceptual data modeling, and tools to support mappings to logical and physical data models in accordance with the three-level data modeling architecture [13].

A mapping from ORM conceptual schemas to hierarchical logical schemas can benefit from a meta model description of a hierarchical logical data model, since a meta model description offers a formal basis for defining the mapping. In this paper, we shall present a meta model description of a hierarchical logical data model. Hierarchical logical data models, as well as a meta model to which they should conform, are only described informally in literature; there is no formal specification of any meta model of a hierarchical logical model that we know of. The most precise (consistent and complete) description of a hierarchical logical model can be found in [14]. Our contribution is to present a formal specification of a meta hierarchical logical data model using ORM. We note that we employ terminology regarding hierarchical data modeling concepts as found in [14].

Before presenting our meta hierarchical logical data model, we introduce a hierarchical logical data model example taken from [14]. The example is a hierarchical logical schema of a company database shown in Fig. 1. This hierarchical logical schema is based upon the ER conceptual schema shown in Fig. 2. Throughout this paper, we will use the example to populate our meta model to validate its correctness.

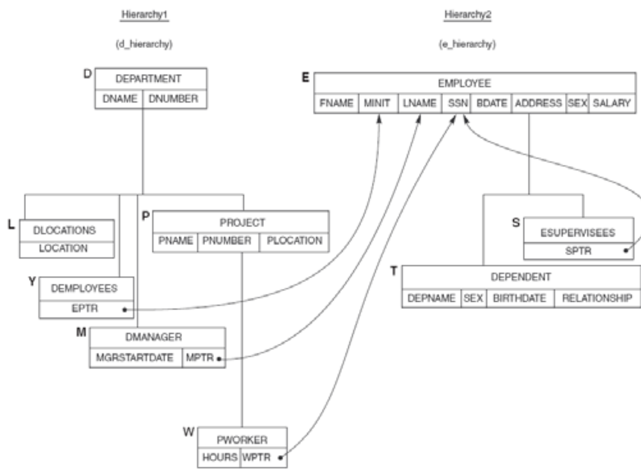


Fig. 1. Hierarchical logical schema diagram of a company database taken from Figure D.8 in [14]. This schema is based upon the ER conceptual schema diagram shown in Fig. 2.

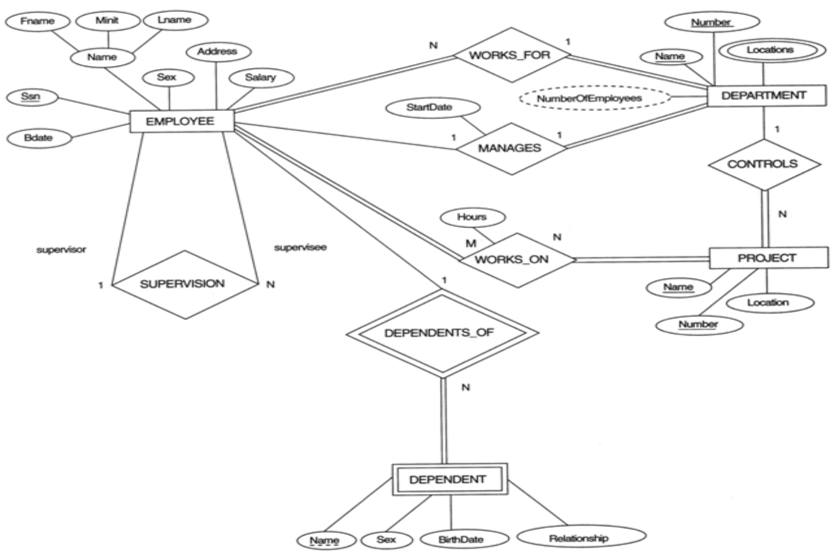


Fig. 2. ER conceptual schema diagram of a company database taken from Figure 7.1 in [14]

The next sections describe the different concepts of our meta hierarchical logical data model. First, the data structure types are introduced, followed by a description of the physical parent-child relationship type concept enforcing a strict physical hierarchy. Subsequently, more advanced concepts like identifiers and logical parent-child relationship types are described. Finally, we offer some conclusions and directions for further research. This paper uses basic ORM notations, cf. [3].

2 Data Structure Types and Strict Physical Hierarchy

In general, a hierarchical logical data model structures data like a tree by means of records, fields and parent-child relationships. Each record represents a real world instance of someone or something about which data are to be collected and stored, e.g. an occurrence of person, place, thing, event or some abstraction. Records that are of the same type are grouped into record types, e.g. all the records representing person instances are grouped into record type ‘Person’. Collected data about each real world instance, e.g. person’s name, phone number and age, are stored within a collection of instantiated fields inside the record representing that instance. The record type to which a record belongs specifies the fields that a record of that type shall instantiate.

Real world relationships between instances are represented by physical parent-child relationships (PCRs). Each physical PCR is a physical relationship between one parent record and zero or more child records. Physical PCRs between parent records of the same type and child records of the same type are grouped into physical parent-child relationship types (PCRTs). The record type on the one-side is called the physical parent record type, and the one on the many-side is called the physical child record type. Each physical PCRT implies a cardinality of at most one-to-many employing a strict physical hierarchy. Hence, each physical PCR of some physical PCRT consists of one record of the physical parent record type and zero or more records of the physical child record type.

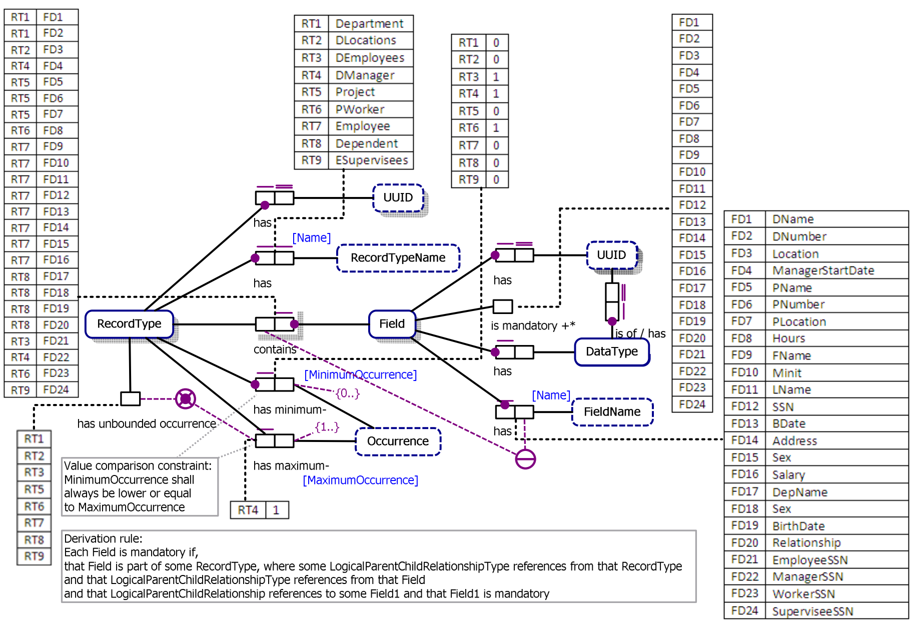


Fig. 3. Formal specification of record types and fields, populated according to the example provided in Fig. 1 and Fig. 2. Note that UUID values are replaced by easier readable values.

Primarily for machine interpretation purposes, Universally Unique Identifiers (UUID) are used to identify record types and fields. For human interpretation purposes, each record type is identifiable by a unique name, and the name of each field is unique within its containing record type. The permitted number of records of a record type participating in a physical PCR with a record of its parent record type, the cardinality, is specified as a combination of minimum, together with maximum or unbounded occurrence. In case a record has no parent record, the default cardinality is zero to unbounded. Each field is either optional or mandatory, indicating that it may or shall contain data. Finally, it is required to specify the data type of the data contained in a field. Note that data types are not specified in more detail. Fig. 3 shows the formal specification of record types and fields.

A record type at the highest level in the hierarchy has no physical parent and is called a root record type. Whether a record type is root, child, parent or a combination is derived. For each record type exactly one of the following holds:

- That record type is a root record type;
- That record type is a root and parent record type;
- That record type is parent and child record type;
- That record type is a child record type.

To enforce a strict physical hierarchy, the physical PCRTs are acyclic and intransitive. The formal specification of the physical PCRT concept is given in Fig. 4.

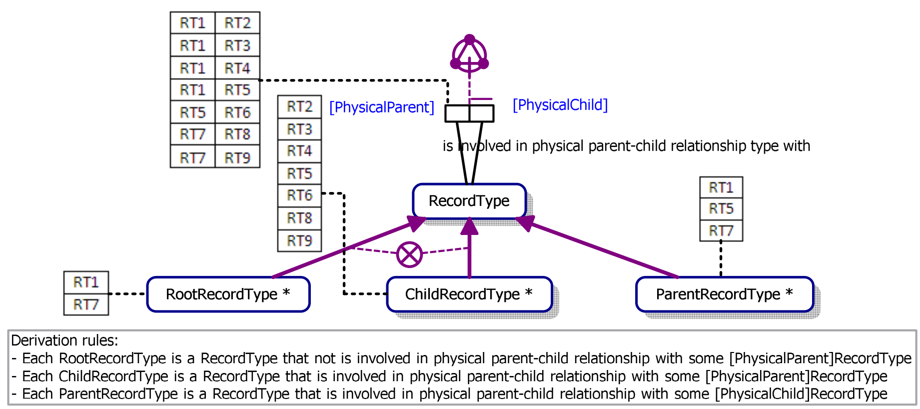


Fig. 4. Formal specification of physical PCRT concept, populated according to the example provided in Fig. 1 and Fig. 2

3 Identifier Concepts

In order to manipulate data stored in a hierarchical model, uniquely identifying each record of a specific record type is required. Our identifier concept, shown in Fig. 5 (left), formally specifies that each record type shall have at least one identifier that provides the means to identify each record of that type, and exactly one of those identifiers provides the preferred identification. Each identifier is uniquely identifiable by a UUID, and has a unique name within the record type it provides identification for.

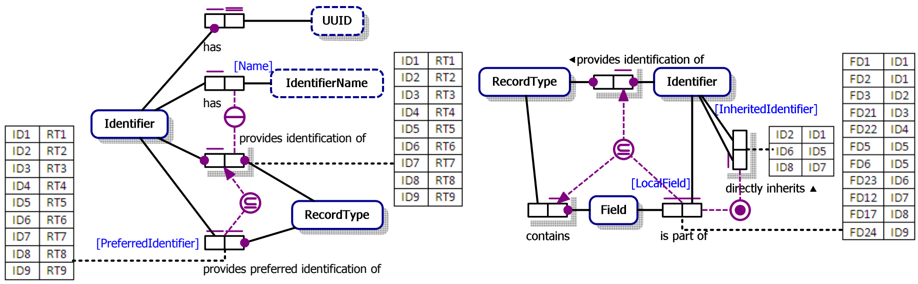


Fig. 5. Formal specification of identifier concept (left), and absolute identifier concept (right), populated according to the example provided in Fig. 1 and Fig. 2

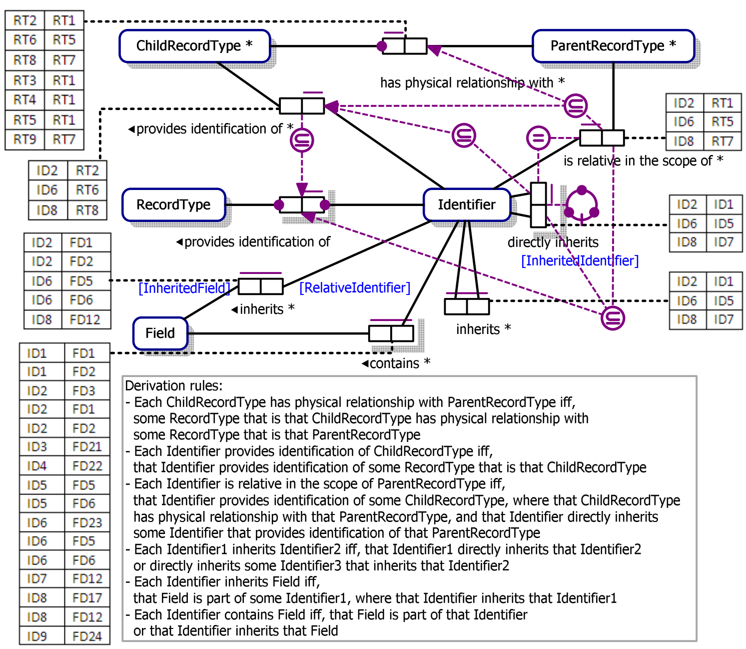


Fig. 6. Formal specification of relative identifier concept, populated according to the example provided in Fig. 1 and Fig. 2

Identifiers in our hierarchical model can be absolute or relative. Our formal specification of an absolute identifier in Fig. 5 (right) specifies that such an identifier shall involve at least one local field, a field of the record type it provides identification for, but does not inherit some identifier. This implies that each record of that record type can occur at most once within the entire hierarchy.

Fig. 5 (right) also shows that a relative identifier may involve at least one local field of the record type it provides identification for. In addition, each relative identifier provides identification of some child record type and shall directly inherit an identifier of its parent record type. As long as an inherited identifier is again relative, the process is repeated recursively until the first absolute identifier of an ancestor is found. Hence, a relative identifier shall always inherit the absolute identifier of an ancestor record type and therefore inherits at least one local field of that ancestor record type. All the other relative identifiers that provide the path to that absolute identifier are also inherited including their local fields. The implication of a relative identifier is that each record of the record type it provides identification for occurs at most once within the scope of a record of the ancestor record type. Fig. 6 presents the formal specification of the relative identifier concept, and Fig. 7 illustrates an example of the identifier concepts.

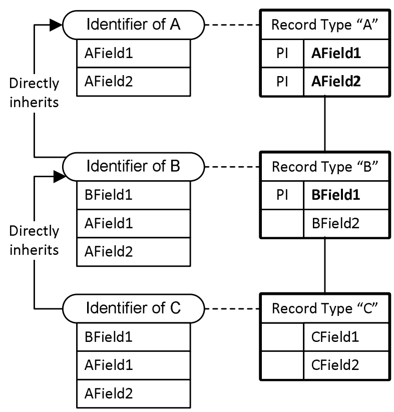


Fig. 7. Example of identifier concepts. On the right a hierarchy of three record types "A", "B" and "C" is shown, with each record type containing two fields where "PI" indicates the fields that are part of the record types' preferred identifier. Each preferred identifier of a record type is shown on the left side, including the local and inherited fields it contains. It clearly shows that the identifiers of B and C are relative, and the identifier of A is absolute.

4 Logical Parent-Child Relationship Type Concept

Many relationships in the real world do not conform to the one-to-many cardinality of the physical PCRTs, implying that a record type has more than one PCRT with the same or additional parent record types. Most concepts to support such relationships involve duplication of data, threatening data integrity.

Logical PCRTs are introduced as a concept to avoid duplication while maintaining data integrity; logical PCRTs cause a departure from the strict physical hierarchy. Logical PCRTs are the same as physical PCRTs except that each logical PCR does not physically relate two records, but instead logically references from one record to another. Logical PCRTs also provide the means to model cyclic structures, by for example allowing a record to reference to itself.

Each record type from which a logical PCRT references is called a logical child record type, and each record type that is being referenced is called a logical parent record type. Referencing a logical parent record type also requires referencing an identifier of that record type. Each logical PCRT relates a set of fields of the logical child record type with an equal set of fields of the referenced identifier identifying the referenced logical parent record type. Each field from which a logical PCRT is referencing, can only be involved in exactly one logical PCRT. Each field to which a logical PCRT is referencing can be referenced by more than one logical PCRT. Finally, each combination of fields related by a logical PCRT is unique.

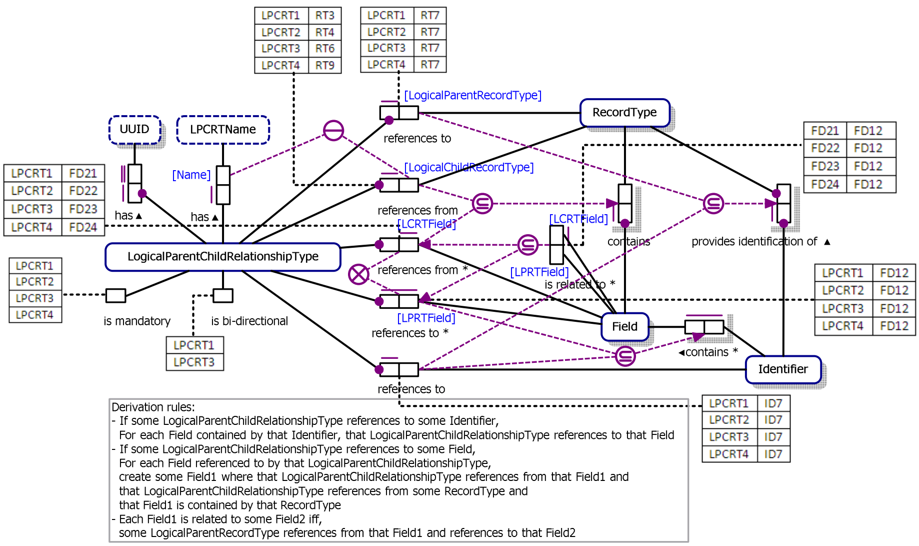


Fig. 8. Formal specification of logical PCRT concept, populated according to the example provided in Fig. 1 and Fig. 2

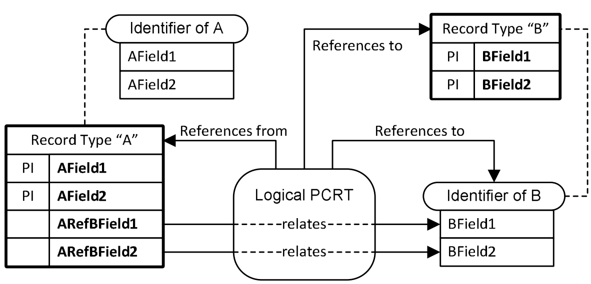


Fig. 9. Example of logical PCRT concept. The logical PCRT references from logical child record type “A” to logical parent record type “B” by referencing the preferred identifier of B. Consequently, the logical PCRT relates two fields of record type “A” with two fields of the identifier of record type “B”.

A logical PCRT can be declared mandatory, indicating that each record of the logical child record type shall participate in a logical PCR with exactly one record of the logical parent record type. In addition, a logical PCRT can be declared bi-directional implying that each record of the logical parent record type shall participate in at least one logical PCR with some record of the logical child record type. Each logical PCRT is uniquely identified by a UUID, and its name is unique within the logical child record type from which it references. The logical PCRT concept is formally specified in Fig. 8. In order to illustrate the logical PCRT concept, an example is given in Fig. 9.

5 Conclusions and Further Research

To the best of our knowledge, we have provided the first formal specification of a meta hierarchical logical data model, which before was described only informally in literature. Our formal specification includes data structure types typical of a hierarchical logical data model; record types, fields, and parent-child relationship types (PCRTs). In addition, our formal specification covers the physical PCRT concept, enforcing a strict physical hierarchy. Furthermore, we have provided formal specifications of more advanced concepts like identifiers and logical PCRTs.

We have also emphasized the importance of having a meta model description of a hierarchical logical data model, when mapping an ORM conceptual schema to a hierarchical logical schema in the context of a three-level data modelling architecture.

Further research aims to extend the meta hierarchical logical data model to support additional constraints, and to formally specify instantiation of the hierarchical logical data model. By using our meta model description we aim to develop a tool including algorithms that map ORM conceptual schemas into hierarchical logical schemas, and hierarchical logical schemas into different types of physical schemas (hierarchical DDL and XSD documents), as well as the reverse mappings.

References

1. Chen, P.P.: The entity-relationship model—towards a unified view of data. ACM Transactions on Database Systems 1(1), 9–36 (1976)
2. Object Management Group 2003, UML 2.0 Superstructure Specification, `http://www.uml.org/`
3. Halpin, T.A., Morgan, T.: Information Modeling and Relational Databases, 2nd edn., pp. 855–856. Morgan Kaufmann Publishers, Burlington (2008)
4. Halpin, T.: Information Modeling in UML and ORM: A Comparison. In: Khosrow-Pour, M. (ed.) Encyclopedia of Information Science and Technology, vol. 3, pp. 1471–1475. Idea Publishing Group, Hershey (2005)
5. Halpin, T.: ORM/NIAM Object-Role Modeling. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Information Systems Architectures, 2nd edn., pp. 81–103. Springer, Heidelberg (2006)
6. Bakema, G., Zwart, J., van der Lek, H.: Fully Communication Oriented Information Modelling, Ten Hagen Stam, The Netherlands (2000)
7. Simsion, G.C., Witt, G.C.: Data Modelling Essentials, 3rd edn., pp. 16–20, p. 235. Morgan Kaufmann, San Francisco (2005)

8. Long, R., Harrington, M., Hain, R., Nicholls, G.: IMS Primer, IBM Redbooks (2000), http://www.redbooks.ibm.com/redbooks/pdfs/sg245352.pdf
9. Bird, L., Goodchild, A., Halpin, T.A.: Object Role Modelling and XML-Schema. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) ER 2000. LNCS, vol. 1920, pp. 309–322. Springer, Heidelberg (2000)
10. Campbell, L.J., Halpin, T.A., Proper, H.A.: Conceptual Schemas with Abstractions - Making flat conceptual schemas more comprehensible. Data & Knowledge Engineering 20(1), 39–85 (1996)
11. Morgan, T.: A Proposal for Folding in ORM Diagrams. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 474–483. Springer, Heidelberg (2010)
12. Vermeir, D.: Semantic Hierarchies and Abstractions in Conceptual Schemata. Information Systems 8(2), 117–124 (1983)
13. Lemmens, I., Sgaramella, F., Valera, S.: Development of Tooling to Support Fact-Oriented Modeling at ESA. In: Meersman, R., Dillon, T., Herrero, P. (eds.) Proceedings of the Confederated International Workshops and Posters on On the Move to Meaningful Internet Systems: ADI, CAMS, EI2N, ISDE, IWSSA, MONET, OnToContent, ODIS, ORM, OTM Academy, SWWS, SEMELS, Beyond SAWSDL, and COMBEK 2009, pp. 714–722. Springer, Heidelberg (2009)
14. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems, 5th edn. Appendix D - An Overview of the Hierarchical Data Model. Addison-Wesley Longman Publishing Co., Inc., Boston (2007)

SWWS 2011, MONET 2011 and SeDeS 2011 PC Co-chairs' Message

The Web has now been in existence for quite some time, and it has produced a major shift in our thinking on the nature and scope of information processing. It is rapidly moving towards an application and knowledge deployment model that requires complex interactions and properly structured underlying semantics. This has resulted in a sudden upsurge of research activity in the problems associated with adding semantics to the Web. This work on semantics will involve data, knowledge, and process semantics. The Semantic Web also plays an essential role in the research areas of mobile technologies, social networking, and mobile services. The application of the Semantic Web to social tools enriches the main functionalities of these tools and emphasizes machine-facilitated understanding of information in order to provide a more productive and intuitive user experience. To discuss such research activities, challenges and solutions, the International IFIP Workshop on Semantic Web and Web Semantics (SWWS 2011), an event organized by the IFIP WG 12.4/2.12 which is in its seventh year, provides a forum for presenting original, unpublished research results and innovative ideas related to the voluminous quantity of research in this area.

There was a very good response to the call for papers, and after a thorough and extensive review process, this year we accepted 24 high-quality papers for presentation at the workshop. Each of these submissions was rigorously peer reviewed by at least three experts. The papers were judged according to their originality, significance to theory and practice, readability, and relevance to workshop topics. Papers selected for publication and presentation at the workshop were organized into a number of tracks, namely:

- ODDI—Ontology Development, Deployment and Interoperability
- DAEC—Data Access and Efficient Computation
- EIPKS—Efficient Information Processing, Exchange and Knowledge Synthesis Algorithms
- MONET—Mobile and Networking Technologies for Social Applications
- SeDeS—Semantic and Decision Support

In previous OTMs some of these were organized as separate workshops. However, to provide a unifying forum, we chose to field these into tracks within the broader framework of SWWS. ODDI discusses the broader issues of ontology development, deployment, Web semantics, and interoperability and gives some important examples of applications. DAEC seeks to define efficient data access and computation algorithms. EIPKS seeks to present algorithms for efficient information processing which will lead to knowledge synthesis. MONET presents new social applications of mobile and networking technologies for serving groups of people "on the move," allowing the sharing of activities and/or interests and collaborating on some tasks in geographically dispersed environments. SeDeS

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 380–381, 2011.
© Springer-Verlag Berlin Heidelberg 2011

presents the latest innovations and achievements of academic/business/governmental communities on decision support systems (DSS).

We would like to express our deepest appreciation to the Program Committee members and external reviewers for their generous contribution of time and effort in maintaining the quality during the review process. We would also like to thank the authors of the submitted papers and all the workshop attendees for their discussions and ideas and for turning the SWWS workshop into a success. Lastly, we would like to thank the members of OTM Organizing Committee, who played an important role in the smooth running of the workshop. We feel that the papers and discussions in the various tracks of SWWS 2011 will further inspire research in the Semantic Web and its applications, especially in the areas of ontology development, ontology evaluation, semantic interoperability, semantic support for human–computer interactions, semantic decision support systems, and mobile and networking technology challenges that appear in social applications.

August 2011

Ernesto Damiani
Elizabeth Chang

Patrizia Grifoni
Fernando Ferri
Irina Kondratova
Arianna D'Ulizia

Yan Tang
Jan Vanthienen

An Ontology-Driven Search Module for Accessing Chronic Pathology Literature

Stephan Kiefer[1], Jochen Rauch[1], Riccardo Albertoni[2], Marco Attene[2], Franca Giannini[2], Simone Marini[2], Luc Schneider[3], Carlos Mesquita[4], and Xin Xing[5]

[1] Home Care / Telemedicine- Fraunhofer Institute for Biomedical Engineering, St.Ingbert, Germany
{stephan.kiefer,jochen.rauch}@ibmt.fraunhofer.de
[2] Institute for Applied Mathematics and Information Technology – C.N.R., Genova, Italy
{riccardo.albertoni,marco.attene, franca.giannini,simone.marini}@ge.imati.cnr.it
[3] Institute for Formal Ontology and Medical Information Science, Saarland University, P.O. Box151150, Saarbrücken, 66041, Germany
luc.schneider@ifomis.uni-saarland.de
[4] Link Consulting, Lisbon, 1000-138, Portugal
carlos.mesquita@link.pt
[5] TZI - Center for Computing and Communication Technologies, Bremen, 28359, Germany
xing@tzi.de

Abstract. This paper presents an advanced search module for bibliography retrieval developed within the CHRONIOUS European IP project. The developed search module is specifically targeted to clinicians and healthcare practitioners searching for documents related to Chronic Obstructive Pulmonary Disease (COPD) and Chronic Kidney Disease (CKD). To this aim, the presented tool exploits two pathology-specific ontologies that allow focused document indexing and retrieval. Besides the search module, an enrichment tool is provided to maintain and to keep up-to date such as ontologies. In addition link with the terms of the MeSH (Medical Subject Heading) thesaurus has been provided to guarantee the coverage with the general certified medical terms and multilingual capabilities.

Keywords: Ontology driven search, Health care literature, Chronic diseases.

1 Introduction

Chronic diseases are mostly characterized by complex causality, multiple risk factors, long latency periods, a prolonged course of illness and functional impairment or disability. Most chronic diseases do not resolve spontaneously, and are generally not cured completely. Chronic diseases may get worse, lead to death, be cured, remain dormant or require continual monitoring. It is then clear that reducing the severity of both the symptoms and the impact would mean significant benefit for both the individual and the society. This is possible in many conditions. Many physicians are very

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 382–391, 2011.
© Springer-Verlag Berlin Heidelberg 2011

optimistic about the benefits that a proper disease management can bring to provide a full and active life and recognize that the long-term health outlook for chronic disease has improved in the past decade, and most of them credit the improvement to better management and monitoring.

To support the management of persons at risk or with chronic health conditions, the project CHRONIOUS (EU Contract N. FP7-ICT-2007–1–216461) is developing an open framework for their remote monitoring and treatment. Chronic obstructive pulmonary disease (COPD) and Chronic kidney disease (CKD) are two case studies to set up a test bed in the project, but the CHRONIOUS architecture is suitable for any chronic disease.

Within this project an intelligent literature Search Module has been developed. The CHRONIOUS Search Module enables healthcare professionals to access well focused and up to date healthcare information in the form of scientific literature, guidelines, hospital-specific documentation and grey literature.

The CHRONIOUS search mechanism overcome Google Scholar, PubMed and GoPubMed by combining the MeSH potentiality with domain ontologies as topic-neutral representations of the domain of entities (objects, processes, qualities, dispositions, functions, etc.) related to the COPD and CKD on ontologies. Indeed, while MeSH encodes terminological knowledge, the ontologies encode expert domain knowledge. In this respect, MeSH and domain ontologies do not compete, but fulfill complementary tasks and the domain ontologies provide additional structure and depth as far as knowledge items are concerned; this additional representational power can be harnessed to improve annotation of and hence search over medical literature.

The Search Module combines two domain ontologies (one for each chronic disease treated) built on top of the Middle Layer Ontology for Clinical Care (MLOCC) [1] with the linguistic capabilities provided by the Medical Subject Heading thesaurus (MeSH) [2].

The use of the ontology for Clinical Care (MLOCC) guarantees the exploitation of well-funded and formalized medical concepts and the integration with other medical domain ontologies (e.g., ACGT Master Ontology [3], Open Biological and Biomedical Ontologies (OBO) foundry [4, 5]). Based on medical guidelines, the two domain-specific ontologies have been defined and properly validated by clinical experts coordinated by an international medical board. Within CHRONIOUS, ontologies and MeSH provide complementary benefits: for each specific disease, indeed, an ontology provides a fine-grained knowledge that MeSH is not supposed to reach; on the other hand, MeSH provides well-established generic medical terminology, multilingual representations, synonyms, narrower/broader/related concepts which can be exploited by the clinicians for the composition of expressive queries in a simple and effective way. The combination of the domain specific ontologies and the MeSH thesaurus allows CHRONIOUS to provide more focused results when compared to traditional search engines for the medical literature. Besides the two domain-specific ontologies, the literature Search Module consists of components for processing and indexing the scientific literature, as well as the components that allow the user to formulate appropriate queries. Moreover, to deal with the problem of medical knowledge evolution, the developed system provides tools for supporting the experts in upgrading the provided ontologies to new emerging concepts in the concerned documentation. This is even more important when considering remote patient monitoring, with wearable sensors, as in CHRONIOUS, since the availability of new signals and information is expected to introduce new practices and new insight for patient care which turns in more evolving concepts.

2 Architecture

The architecture of the CHRONIOUS Search Module, depicted in Fig. 1, is based on different sub-modules including:

Upload Tools to load documents related to CKD and COPD in the CHONIOUS System for their indexing. It provides two different capabilities: one supporting the clinicians to upload their own documents, the second offering capabilities to automatically treat resources directly coming from publishers upon specific agreements;

Transformation Module converts the imported documents, which might be in different formats, to normalized plain text format;

Natural Language Processing (NLP) Tool is based on the GATE Framework [6], allows to specify processing pipelines (e.g. consisting of Sentence Splitter, Tokeniser, Part-of-speech Tagger and Morphological Analyser) as well as filter parameters (e.g. for word kinds and categories). The NLP module is also exploited by the Ontology Enrichment tool for the extraction of candidate concepts from the documents;

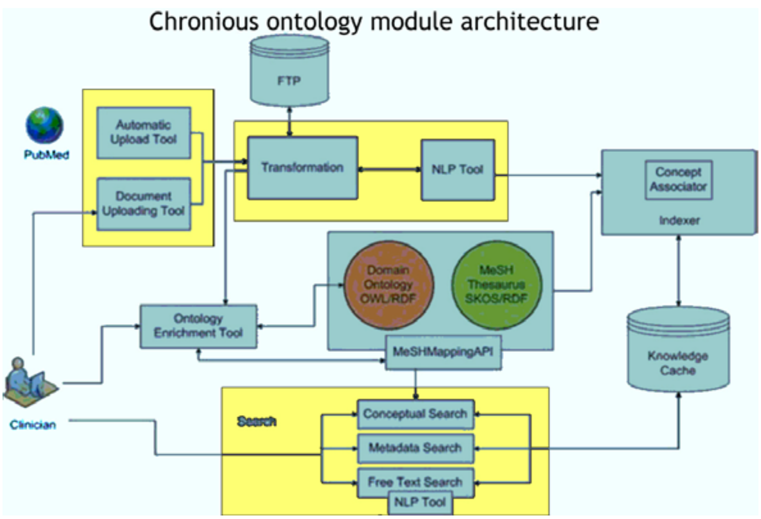


Fig. 1. The CHRONIOUS Search Module components

Knowledge Cache maintains a graph database to store the generated indices and connections;

Search Module offers search methods as well as several tools that facilitate the user's interaction with the system;

Ontology/Thesaurus module incorporates the domain knowledge. It includes the CHRONIOUS Ontologies, MeSH and its translations in different languages, as well as the mapping between MeSH and Ontologies;

Ontology Enrichment module is responsible for the maintenance of the ontology in a semi-automatic way. It uses NLP tool to pre-process the imported documents and to find word groups and linguistic patterns that are related to the domain.

In the following paragraphs, the most important modules are detailed.

2.1 Natural Language Processing Tool

Within the CHRONIOUS Search Module, the NLP tool is applied to the text documents provided by the transformation component to automatically extract structured information. The specific annotations provided by the NLP tool are used by the Document Indexer and Concept Associator as well as by the Ontology Enrichment Tool for proposing new concepts. The NLP Tool runs on a server and can be accessed through a web service. Transformation tool provides methods for receiving documents in different formats and responds with processed data. These operations also allow defining filter parameter (for annotation types and features) in order to focus on the actual required data and to reduce the number of delivered annotations.

For implementing the NLP algorithms the GATE framework [6] has been used that allows developing and deploying software components that process human language. The NLP Tool extracts the headwords of a text using different modules: the *Tokenizer* splits the sentences of the text obtained by the *Sentence Splitter* into very simple tokens such as numbers, punctuation and words of different types; the Part-of-speech (POS) Tagger produces part-of-speech tags (e.g. noun, proper noun, preposition) as an annotation on each word or symbol. Finally the *Morphological analyser* identifies the lemma (headword) and affix (e.g. “runs”, “ran” and “running” has all “run” as lemma) by considering one token and its part-of-speech tag, one at a time.

In addition to the above standard processing sequence, in order to satisfy the requirements of the Ontology Enrichment Tool and the Indexer, in the NLP Tool some additional resources of the GATE framework have been integrated:

- **OntoRoot Gazetteer [7]:** A GATE plugin that produces ontology-aware annotations for extracted terms (ontology matching using names and labels of the ontology concepts).
- **Shallow Parser:** analyzes the sentences to identify word groups by linguistic patterns (e.g. “chronic disease”, “lung function”). For the prototype implementation, which is focused on the English language, we extracted and utilized a JAPE transducer, executes a linguistic pre-processing in terms of ontology learning, from the Text2Onto project [8].
- **RegEx-Pattern Matcher:** matches the lemma of a token with word patterns defined as regular expressions. It has been implemented as JAPE resource, which expects an input list with entries coded as regular expressions.
- **Dictionary Matcher:** matches the lemma of a token to a (common) dictionary. In our implementation it uses the WordNet dictionary [9] and Java API for searching.

- **Thesaurus Matcher:** matches the lemma of a token to a (domain) thesaurus. For the target medical domain we have implemented a JAPE resource, which uses the MeSH-Mapping API to access MeSH terms and the mapped ontology concepts.

2.2 Search

To better support users various search methods have been provided together with tools that facilitate the interaction with the system and with functionalities for the ontology/thesaurus browsing and the query building. Search is then possible as:

- **Metadata Search,** which is a keyword-based search looking into the metadata information associated with a document and indexed in the Knowledge Cache.
- **Conceptual Search,** which allows the user to submit queries at a higher level of abstraction as compared to keyword based search by retrieving content that does not contain keywords from the query. The conceptual search consist of three components the user can exploit to build and refine their queries:
  - An Ontology Browser panel displaying a clinical view, which is a simplified view of the ontology structure enabling the user to exploit relations with connected concepts.
  - A Concept Finder tool that provides on-type suggestions of concepts included in the Ontology and/or MeSH and that exploits the multilingual capabilities of MeSH.
  - A Query Building tool where queries are build combining Ontology and MeSH concepts through boolean expressions.
- Free-text search: this is also a conceptual-based search allowing the user to submit queries in natural language by exploiting the NLP capabilities. The NLP Tool processes the sentence, keeps the valuable information and the search investigates the associations stored in the Knowledge Cache. This search exploits the multilingual capability of MeSH allowing the user to use concept terms in their languages (currently in Italian, Spanish and Portuguese).

To avoid copyright infringement, while guaranteeing a sufficient coverage of the search capabilities, full papers are used only for indexing purpose. The search is performed on the metadata and indexed concepts, and displays the results' metadata which also include for published document the related DOI. Therefore access to the document is performed by dereferencing the HTTP URI of the paper's DOI respecting the paper access constraints and users' subscriptions. The solution has been verified to be acceptable for getting access to the bibliography provided by the main publishers.

2.3 Ontology/Thesaurus Module

This module provides the access to all the domain and linguistic resources used in the annotation and retrieval of medical reference documents enhancing the search functionalities therewith.

Through an ad hoc developed API, the Ontology and thesaurus module provides access to: COPD [13], CKD [14] and MLOCC (middle-layer) [15] ontologies; MeSH (English) [2] thesaurus version 2010; MeSH multilingual lexicographic representations (currently the Italian, the Spanish and the Portuguese versions have been provided); mapping between MeSH concepts and correspondent Ontologies classes.

The CHRONIOUS ontologies encode expert knowledge relevant to the envisaged domain (COPD and CKD) such as to provide a rich science-driven repository of concepts for annotating and searching medical literature in the scope of the project, as well as a wealth of relations to refine domain-specific literature search. The COPD and CKD ontologies are built on top of a Middle Layer Ontology for Clinical Care (MLOCC) whose purpose is to provide a link between the leaves of Basic Formal Ontology [12], a foundational ontology used in the Open Biological and Biomedical Ontologies (OBO) Foundry [5], and the top-nodes of the domain ontologies. MLOCC represents, as it were, the common core of biomedical and clinical knowledge shared by the COPD and CKD Ontologies. Furthermore, we have expanded the ontology providing reference for selected terms from MeSH. MeSH is a certified and very rich thesaurus, that covers a large part of the concepts pertaining to the medical domains. It is already used to index medical articles, but

- it does not deepen specifically the two diseases considered in the project, so it lacks of the required specificity to support in complex search pertaining the considered diseases;
- it is not compliant with the newest web standard suggested by the W3C, so it cannot be easily combined with the OWL [11] domain ontologies.

In the project, the former limitations are overcome by ontologies about COPD and CKD. However, searching for scientific papers may require to move from concepts strictly related to the aforementioned diseases to concepts less specific and vice versa, thus some connections between the two diseases-specific ontologies and the MeSH concepts are required. For this reason, a MeSH-Ontologies mapping linking the concepts provided by MeSH to the ontologies classes has been provided.

To ensure that the thesauri and COPD/CKD ontologies can be fruitfully connected, we have SKOSified MeSH and their Multilingual versions: we have mapped\encoded the MeSH content into Simple Knowledge Organization System (SKOS) model following the rules discussed in [16] and serialized the SKOS in RDF. SKOSyfied MeSH thesaurus, its translations and its mapping to concept described in the Diseases Ontologies are accessible by API developed in CHONIOUS which relies on the JENA framework [10]. Ontologies are accessible through standard OWL 2 API [17].

2.4 Ontology Enrichment Tool

As mentioned, since in the project we aim at supporting care givers in remote monitoring and managing chronic patients, for which knowledge and literature are still at their infancy, it is crucial to have tools for helping the upgrade of the ontology when new concepts arise. Therefore, we decided to exploit the possibility of deriving new concepts from pertinent literature in a semi-automatic way. Thus, the so-called ontology enrichment tool has been developed allowing the identification of gaps in the ontologies with respect to newly inserted documents to support the ontology curator. In particular, the tool combines ontology learning techniques and submission functionalities. Methods for ontology learning are based on linguistic, statistical and machine learning approaches. Thus the Ontology Enrichment Tool uses the NLP Tool described in section 2.1 for a linguistic processing of the transformed document data and annotate the text with several features. Based on these annotations candidate concepts are extracted in a second step. After the pre-linguistic processing the extracted candidate concepts are rated concerning their relevance by points based on several criteria:

- **Corpus relevance:** the relevance of each candidate concept is determined by computing its average Term Frequency Inverted Document Frequency (TF.IDF) value with respect to the whole document corpus.
- **Domain relevance:** matchings with a common dictionary, a domain thesaurus and with regular expression patterns is used to determinate the domain relevance of a candidate concept. The assignment of synonyms results in a higher relevance. This means that the candidate concept has been identified as a synonym of an existing concept (by dictionary synonyms or ontology mapped MeSH thesaurus terms).
- **Subclass-of relations:** the relevance of the candidate concept is raised, if it is possible to assign a subclass-of relation to it. Such an assignment can be determined by the extraction of vertical relations, linguistic patterns or dictionary hypernyms.
- **Concept co-occurrences:** the co-occurrences of the candidate concept with concepts in sentences can be considered as an indicator of a possible relation extraction. The average distance in the text between the candidate concept and a concept within the corpus is calculated as a benchmark.

Candidate extensions need to be validated by clinicians which are supported by the submission facility of the Ontology Enrichment Tool, i.e. a GUI that allows the user review the extracted data. Furthermore the user has the possibility to administrate the candidate concepts in terms of a workflow state ("new", "to validate", "postponed", "accepted" or "rejected"). In this way the submission function of the Enrichment Tool provides suggestions for new concepts (possibly with subclass-of relation assignment) or new labels of existing concepts, but modifications of the ontology must still be done by the ontology expert with an external tool (e.g. Protégé).

3 Evaluation

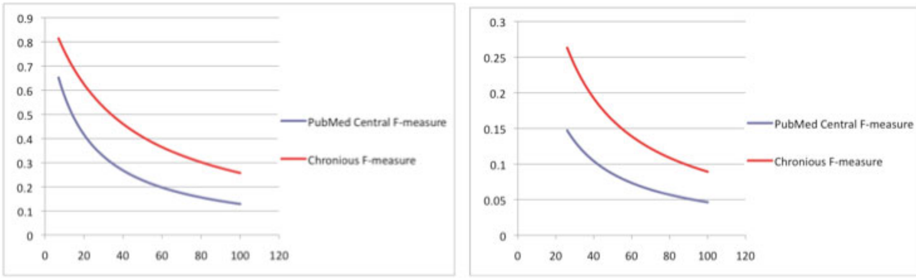
Building domain-specific ontology requires large involvement of experts and close cooperation with technical partners. Both the CHRONIOUS ontologies developed by computer engineers and the mapping relations between MeSH concepts and ontology classes have been evaluated by medical experts. Questionnaires checking the agreement among medical experts on the correctness of the developed ontologies and of the MeSH mapping relations have been created by test engineers and filled in by COPD and CKD clinicians. The evaluation result shows that most of the ontology classes have been defined correctly, a few ontology classes needed to be modified (eight of 964 COPD classes and nine of 972 CKD classes), and together 150 new classes needed to be added (80 new classes for COPD ontology and 70 for CKD ontology). That is less than 7.8% of the total number of defined ontology classes. Furthermore, 42 of 120 mapping relations between MeSH concepts and COPD ontology classes, as well as 43 of 138 mapping relations between MeSH concepts and CKD ontology classes have been modified. The CHRONIOUS search mechanism overcome Google

Scholar, PubMed and GoPubMed by combining the MeSH potentiality with the two specific domain ontologies. Indeed Google Scholar supports the search of documents by providing a simple search by keyword mechanism. Such a mechanism is based on a pure syntactical search of the documents, without taking into account any semantic interpretation of the proposed query. As a consequence, two queries having different syntax but equivalent semantics yield different search results. PubMed improves Google Scholar by exploiting the synonymous provided by MeSH, where the thesaurus is used to produce a semantic interpretation of the query keyword mechanism. The GoPubMed search engine, improves PubMed by considering the concepts of Gene ontology (GO) during the query refinement step. In this case, the clinicians are facilitated during the query composition by exploiting both the terms of MeSH and the terms carried by the Gene ontology. Neither PubMed nor GoPubMed are supported by the specific knowledge related to the specific domain of the COPD and CKD ontology that provide additional structure and depth as far as knowledge items are concerned; this additional representational power is exploited to improve annotation of and hence search over medical literature.

In order to complete the previous qualitative comparison of the CHRONIOUS Conceptual Search, a quantitative evaluation of the performance of the system, i.e., the exactness and completeness of its search result, has been evaluated. Exactness (or *Precision*) measures the number of correctly found documents – documents which are relevant to the search query – as a percentage of the number of documents found, whereas completeness (or *Recall*) measures the number of correctly found documents as a percentage of the total number of existing relevant documents. Usually, Precision and Recall scores are not discussed in isolation. Instead, both are combined into a single measure – the F-measure [18], which is the weighted harmonic mean value of Precision and Recall. To quantitatively evaluate the search performance of the CHRONIOUS Conceptual Search, seven commonly used COPD terms and five CKD terms have been defined by medical experts as search queries. The F-measure of search results of each search query with the CHRONIOUS Conceptual Search has been calculated and compared with one of the most popular text-based search engines – PubMed Central. To have meaningful comparison of the two systems we executed the test with same document database. Thus, documents meeting with certain limitations (e.g., published in a specific year, contain specific terms in their text body, open access, etc.) have been downloaded from PubMed Central and uploaded into the CHRONIOUS document repository. The evaluation result shows that the overall search performance of CHRONIOUS Conceptual Search with defined search queries is in most cases better than PubMed Central. Fig 2 shows the F-measure comparison results with two search terms: “Inhaler Device” and “PostBronchodilator Spirometry”.[1] Furthermore, the overall end user satisfaction of the CHRONIOUS Search Module has been investigated with the medical experts by a questionnaire. The evaluation has shown that 90% of them generally accept the Conceptual Search development and 60% of them gave positive evaluation (“Somewhat satisfied”) to the Free-text Search option.

---

[1] The search results of COPD terms have been retrieved from 930 documents that were published from January 1st to December 31st, 2008 and contained the term “COPD” in their text (incl. title, abstract, text body, figure/table caption, etc.).



(a) (b)

Fig. 2. F-measure comparison between CHRONIOUS Conceptual Search and PubMed Central with search query (a) "Inhaler Device" and (b) "PostBronchodilator Spirometry". The X-axis presents the estimated total number of existing relevant documents in the repository for the corresponding search query (increased incrementally).

4 Conclusions

In this paper, we presented a literature search module developed within the FP7 IP Project CHRONIOUS to complement out of shell search modules such as Google Scholar, PubMed and GoPubmed by improving the search capabilities within the COPD and CKD pathologies, and by providing the possibility to index and retrieve hospital-specific internal documentation. The adopted ontology-thesaurus approach aims at overcoming limitations of pure syntactical document search, in which two queries having different syntax but equivalent semantics yield different search results. The search mechanism combines MeSH terminological power with formal knowledge of the domains specified by the ontologies. MeSH and domain ontologies do not compete, but fulfill complementary tasks. MeSH provides well-established generic medical terminology, multilingual lexical representations and synonyms while domain ontologies provide additional representational power can be harnessed to improve medical literature annotation and retrieval.

Acknowledgements. This research has been supported by the IP project CHRONIOUS (EU Contract N. FP7-ICT-2007–1–216461). The authors want to thanks all the partners for their support. This paper was written as a joint contribution; authors are listed in alphabetical order grouped by their affiliations, besides they have equally contributed to this paper.

References

1. Schneider, L., Brochhausen, M.: The CHRONIOUS Ontology Suite: Methodology and Design Principles. In: Proceedings of the International Conference on Biomedical Ontology, July 26-30. University at Buffalo, NY (2011)
2. Medical Subject Headings (MeSH), http://www.nlm.nih.gov/mesh

3. Brochhausen, M., Spear, A.D., Cocos, C., Weiler, G., Martin, V., Anguita, A., Stenzhorn, H., Daskalaki, E., Schera, F., Schwarz, U., Sfakianakis, S., Kiefer, S., Doerr, M., Graf, N., Tsiknakis, M.: The ACGT Master Ontology and Its Applications - Towards an Ontology-Driven Cancer Research and Management System. Journal of Biomedical Informatics, E-published ahead of print (2010), doi:10.1016/j.jbi, 04.008
4. Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C.J., Neuhaus, F., Rector, A., Rosse, C.: Relations in Biomedical Ontologies. Genome Biology 6, R46 (2005)
5. The Open Biological and Biomedical Ontologies Foundry, `http://www.obofoundry.org/`
6. The GATE framework, `http://gate.ac.uk/`
7. OntoRoot Gazetteer, `http://gate.ac.uk/sale/tao/splitch13.html#sec:gazetteers:ontoRootGaz`
8. Text2Onto project, `http://code.google.com/p/text2onto/`
9. WordNet, `http://wordnet.princeton.edu/`
10. JENA – A Semantic Web Framework for Java, `http://jena.sourceforge.net/`
11. OWL 2 Web Ontology Language Primer W3C Recommendation, October 27 (2009), `http://www.w3.org/TR/owl2-primer/`
12. Spear, A.: Ontology for the Twenty First Century: An Introduction with Recommendations, BFO manual (2006)
13. Chronic obstructive pulmonary disease (COPD) ontology, `http://www.ifomis.org/chronious/copd`
14. Chronic kidney disease (CKD) ontology, `http://www.ifomis.org/chronious/ckd`
15. Middle Layer Ontology for Clinical Care (MLOCC), `http://www.ifomis.org/chronious/mlocc`
16. Van Assem, M., Malaisé, V., Miles, A., Schreiber, G.: A Method to Convert Thesauri to SKOS. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 95–109. Springer, Heidelberg (2006)
17. The OWL API, `http://owlapi.sourceforge.net/`
18. van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworth (1979)

KD2R: A Key Discovery Method for Semantic Reference Reconciliation

Danai Symeonidou, Nathalie Pernelle, and Fatiha Saïs

LRI (CNRS & Paris-Sud XI University)/INRIA Saclay,
LRI, Bât 650 UniversitéParis-Sud 11, 91405 Orsay, Cedex, France
Danai.Symeonidou@inria.fr,
{Nathalie.Pernelle,Fatiha.Sais}@lri.fr

Abstract. The reference reconciliation problem consists of deciding whether different identifiers refer to the same world entity. Some existing reference reconciliation approaches use key constraints to infer reconciliation decisions. In the context of the Linked Open Data, this knowledge is not available. We propose KD2R, a method which allows automatic discovery of key constraints associated to OWL2 classes. These keys are discovered from RDF data which can be incomplete. The proposed algorithm allows this discovery without having to scan all the data. KD2R has been tested on data sets of the international contest OAEI and obtains promising results.

1 Introduction

The reference reconciliation problem tries to find whether different references refer to the same entity (e.g. the same restaurant, the same gene, etc.). There are a lot of approaches (see [3] or [10] for a survey) that aim to reconcile data. Recent global approaches exploit the existing dependencies between reference reconciliation decisions [7,2,1]. In such approaches, the reconciliation of one reference pair may entail the reconciliation of another reference pair. A knowledge based approach is an approach in which an expert is required to declare knowledge that will be used by the reference reconciliation system [5,2]. Some approaches such as [5] use reconciliation rules that are given by an expert, while other approaches such as [7] use the (inverse) functional properties (or the keys) that are declared in the ontology. Nevertheless, when the ontology represents many concepts and when data are numerous, such keys are not easy to model for the ontology expert.

The problems of key discovery in OWL ontologies and key discovery or Functional Dependency discovery in relational databases are very similar. In the relational context, key discovery is a sub-problem of extracting functional dependencies (FDs) from the data. [9] proposes a way of retrieving probabilistic FDs from a set of data sources. Two strategies have been proposed: the first one merges the data before discovering FDs while the second one merges the FDs obtained from each data source. This paper focuses on the problem of finding probabilistic FDs with only a single attribute in each side.In order to find the FDs, TANE [4] partitions the tuples into groups based on their attribute values. The goal is to find approximate functional dependencies: functional

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 392–401, 2011.
© Springer-Verlag Berlin Heidelberg 2011

dependencies that almost hold. In the context of Open Likned Data, [6] have proposed a supervised approach to learn functional dependencies on a set of reconciled data.

There are a lot of works that deal with the discovery of FDs in relational databases, however only a few of them focus on the specific problem of retrieving keys. The Gordian method [8] allows discovering composite keys in relational databases. In order to avoid to checking all the possible combinations of candidate keys, the method proposes the discovery of the non-keys in a dataset and then using them to find the keys. In this method a prefix tree is built and explored (using a merge step) in order to find the maximal non keys. To optimize the tree exploration, they exploit the anti-monotone property of a non key. Nevertheless, it is assumed that the data are completely described (without null values). Furthermore, multivalued attributes are not taken into account.

In this paper, we present KD2R which is an automatic approach for key discovery in RDF data sources which conform to the same (or aligned) OWL2 ontology (the ontology can be represented in RDFS or in OWL). These keys are discovered from data sources where the UNA (Unique Name Assumption) is fulfilled and the information can be incomplete and multi-valued. To avoid scanning the whole data source, KD2R discovers first maximal non keys before inferring the keys. KD2R exploits key inheritance between classes in order to prune the non key search space. KD2R approach has been implemented and evaluated on two different data sources.

The paper is organized as follows: in section 2, we describe the data and the ontology model. In section 3, we present the KD2R and then we present first experiment results in section 4. Finally, in section 5 we conclude and give some future work plans.

2 Reference Reconciliation Based on Key Constraints

2.1 Ontology and Data Model

Data are represented in RDF–Resource Description Framework–(www.w3.org/RDF). For example, the RDF source S1 contains the RDF descriptions of four museums in the form of a set of class facts and property facts (relational notation):

Source S1:
$ArchaeologicalMuseum(S1\_m1), museumName(S1\_m1, Archaeological\ Museum),$
$located(S1\_m1, S1\_c1), museumAddress(S1\_m1, 44\ Patission\ Street),$
$inCountry(S1\_m1, Greece), Museum(S1\_m2), museumName(S1\_m2,$
$Centre\ Pompidou), contains(S1\_m2, S1_p4), contains(S1\_m1, S1\_p5),$
$museumAddress(S1\_m2, 19\ rue\ Beaubourg), inCountry(S1\_m2, France),$
$Museum(S1\_m3), museumName(S1\_m3, Musee\ d'orsay),$
$museumAddress(S1\_m3, 62\ rue\ de\ Lille), inCountry(S1\_m3, France)$
$WaxMuseum(S1\_m4), museumName(S1\_m4, Madame\ Tussauds), located(S1\_m4,$
$S1\_c4), museumAddress(S1\_m4, Marylebone\ Road), inCountry(S1\_m4, England)$

The examined RDF data are in conformity with a domain Ontology represented in OWL2 (http://www.w3.org/TR/owl2-overview). The OWL 2 Web Ontology Language provides classes, (data or object) properties, individuals and data values. In the Museum ontology (see Figure 1), the class $Museum$ is described by its address (*owl:DataProperty* $museumAddress$), its location (*owl:ObjectProperty* $located$),

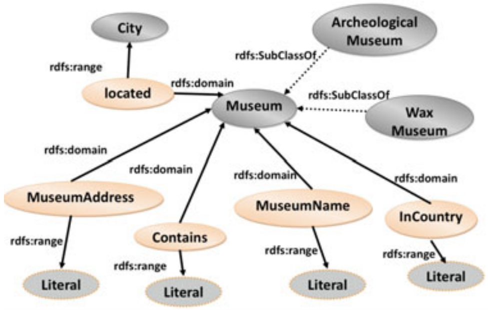


Fig. 1. Museum Ontology

its name (*owl:DataProperty* $museumName$) and its country (*owl:DataProperty* $inCountry$). The classes $ArcheologicalMuseum$ and $WaxMuseum$ are subsumed by the class $Museum$.

In OWL2, it is possible to express key axioms for a given class: a key axiom HasKey (CE (OPE1 ... OPEm) (DPE1 ... DPEn)) states that each (named) instance of the class expression CE is uniquely identified by the object property expressions OPEi and by the data property expressions DPEj. This means that no two distinct (named) instances of CE can coincide on the values of all object property expressions OPEi and all data property expressions DPEj. An ObjectPropertyExpression is either an ObjectProperty or InverseObjectProperty. A data property expression is an *owl:DataProperty*.

For example, we can express that the object property $located$ is a key for the class $City$ using $HasKey(kd2r : City(inverse(kd2r : museumAddress))())$

2.2 Constraint Integration in Reference Reconciliation

LN2R [7] is a logical (L2R) and a numerical (N2R) method for reference reconciliation. L2R and N2R use the knowledge given in a OWL (or OWL2) ontology to reconcile data. L2R translates keys, disjunctions between classes and the Unique Name Assumption (UNA) [1] into reconciliation rules. These rules infer both (non) reconciliations facts and synonyms for literal values. For example, since $located$ is a key for the $City$ class (one museum is located in only one city) the following rule is generated by L2R:

$$City(L1) \wedge City(L2) \wedge Reconcile(X, Y) \wedge Located(X, L1) \wedge Located(Y, L2) \implies Reconcile(L1, L2)$$

A logical reasoning based on the unit-resolution inference rule is used to infer all the (non) reconciliations.

N2R exploits keys in order to generate a similarity function that computes similarity scores for pairs of references. This numerical approach is based on equations that model the influence between similarities. In the equations, each variable represents the (unknown) similarity between two references while the similarities between values of data properties are constants The functions modelling the influence between similarities are a combination of the maximum and the average functions in order to take into account the keys declared in the OWL ontology in an appropriate way (see [7] for more details).

[1] UNA declares that all the references that appear in a source cannot be reconciled.

2.3 Key Discovery Problem Statement

When RDF data are numerous, heterogeneous and published in clouds of data, the keys that are needed for the reconciliation step are not often available and cannot be easily specified by a human expert. Therefore, we need methods to discover them automatically from data. The key discovery has to face several kinds of problems, due to data heterogeneity: absence of UNA, syntactic variations in data, erroneous values and incompleteness of information. When UNA is not fulfilled, we cannot distinguish between the two cases: (i) two equal property values describing two references which refer to the same real world entity and (ii) two equal property values describing two references which refer to two distinct real world entities. This ambiguity leads to missing keys that can be discovered. In RDF data each instance of a class can be described by a subset of properties that are declared in the ontology. The incompleteness of data entails the discovery of keys that may be incorrect.

In this paper we focus on the problem of key discovery in RDF data when UNA assumption is declared for each data source and where there are no erroneous values.

3 KD2R: Key Discovery Method for Reference Reconciliation

KD2R method aims to discover keys as exact as possible, with respect to a given dataset in order to enrich a possible existing key set.

The most naive automatic way to discover the keys is to check all the possible combinations of properties that refer to a class. The keys should uniquely identify each instance of a class. Let us assume that we have a class which is described by 15 properties in order to estimate the cost of this naive way. In this case the number of candidate keys is $2^{15}-1$. In order to minimize the number of computations as much as possible we have proposed a method inspired from [8] which first retrieves the set of maximal non keys and then computes the set of minimal keys, using this set of non keys. Indeed, to make sure that a set of properties is a key we have to scan the whole set of instances of a given class. On the contrary, finding two instances that share the same values for the considered set of properties would suffice to be sure that this set of properties is a non-key.

We present, first, how we have defined non keys, keys and undetermined keys for a class in a given RDF data source and for a given set of RDF data sources. Then we will present the KD2R-algorithm that is used to find keys for the ontology classes.

3.1 Keys, Non Keys and Undetermined Keys

Let S be a data source for which the UNA is declared, and P_c be the set of RDF properties defined for a class C of the ontology O.

Definition 1 (Non keys). A set of property expressions $nk_{CSi} = \{pe_1, \ldots, pe_n\}$ is a non key for the class C in S if:

$\exists X \in S, \exists Y \in S$ s.t. $(C(X) \wedge C(Y) \wedge pe_1(X, a_1)) \wedge pe_1(Y, a_1) \wedge \ldots \wedge pe_n(X, a_n)) \wedge pe_n(Y, a_n)) \wedge X \neq Y)$

We denote NK_{CS} the set of non keys $\{nk_{CS1}, ..., nk_{CSm}\}$ of the class C w.r.t the data source S.

Definition 2 (Keys). A set of property expressions $k_{CSi} = \{pe_1, \ldots, pe_n\}$ is a key for the class C in S if:

$$\forall X \in S, \forall Y \in S\,(C(X) \wedge C(Y)) \rightarrow (\exists pe_j \in k_{CSi} \text{ s.t. } pe_j(X,a)) \wedge pe_j(Y,b)) \wedge a \neq b)$$

We denote K_{CS} the key set $\{k_{CS1}, ..., k_{CSm2}\}$ of the class C w.r.t the data source S.

Definition 3 (Undetermined Keys). A set of property expressions $uk_{CSi} = \{pe_1, \ldots, pe_n\}$ is an undetermined key for the class C in S if: (i) $uk_{CSi} \notin NK_{CS}$ and (ii) $\exists X \in S, \exists Y \in S$ s.t. $((C(X) \wedge C(Y) \wedge \forall pe_j \in uk_{CSi}(pe_j(X,a) \wedge pe_j(Y,b) \implies a = b) \wedge \exists pe_w \in uk_{CSi}$ s.t.$(\nexists pe_w(X,Z) \vee \nexists pe_w(Y,V))$

For example, $\{InCountry, Located\}$ is an undetermined key, since there are two museums in the same country but one of the cities is unknown. Hence, we cannot decide if it represents a key or a non-key.

We denote UK_{CS} the set of keys $\{uk_{CS1}, ..., uk_{CSm3}\}$ of the class C w.r.t the data source S.

Keys for a given set of data sources. Let $\mathcal{S} = \{S1, S2, \ldots, Sm\}$ be a set of m data sources for which the UNA is declared. Let $K_{CS1}, \ldots, K_{CSm}$ be the respective set of keys of $S1, S2, \ldots, Sm$, the set of keys $Kc_{\mathcal{S}}$ that is satisfied in all the sources is the set of minimal keys that belong to the Cartesian product of $K_{CS1}, \ldots, K_{CSm}$.

3.2 KD2R Algorithm

Given a set of datasets and a domain ontology, KD2R-algorithm allows to find keys for each instantiated class. It follows a top-down computation in the sense that the keys that are discovered for a class are inherited by its sub-classes. KD2R uses a compact representation of RDF data expressed in a prefix-tree in order to compute the complete set of maximal undermined keys and maximal non keys and then the complete set of minimal keys.

Prefix-Tree creation. In this section we will present the creation of the prefix-tree which represents the RDF descriptions of a given class.

As it is illustrated in Figure 2, each level of the tree corresponds to an instantiated property expression. Each node contains a variable number of cells. Each cell contains the distinct data property values or the distinct URIs of the object property expression of the considered level. Each cell contains the list of URIs referring to the corresponding class instances. Each non-leaf cell has a pointer to a single child node. Each unique prefix path represents the set of instances that share one value or one URI for all the properties involved in the path.

For the sake of simplicity we will use the term $value$ to either refer to basic values of data properties or to URIs of object properties.

In order to represent the cases where property values are not given (i.e. null values in relational databases) we create first an intermediate prefix-tree. In this intermediate prefix-tree, an artificial null value is created for those properties. Then, the final prefix-tree is generated by assigning the set of all the possible values to each artificial null value, i.e. those existing in the dataset.

Intermediate Prefix-Tree creation. In order to create the intermediate prefix-tree we use the set of all properties that appear at least in one instance of the considered class. For each instance, for each property and for each value if there is no cell which already contains the property value a new cell is created. Otherwise, the cell is updated by adding the instance URI to its associated list of URIs. When a property does not appear in the source, we create or update, in the same way, a cell with an artificial null value. Let it be noted that the intermediate prefix-tree creation is done by scanning the data only once.

Final Prefix-Tree creation. The final prefix-tree is generated from the intermediate prefix-tree by assigning the set of the possible values contained in the cells of this node to each artificial null value of a given node, if it exists. More precisely, the null cell is deleted and its URI list is added to all the other cells of the node. Then, for the descendants of this node, we recursively apply the processing of artificial null values and the node merge operation which is described in the following.

Node merge operation. This operation takes the list of nodes that need to be merged as input and provides a merged node which contains one cell per distinct value that exists in the input list of nodes as output. The new URI list of each cell contains all the URI lists of the merged cells. This merge operation is performed recursively for all the descendants of the considered nodes.

In figure 2, we give the final prefix-tree of the RDF data described in section 2.

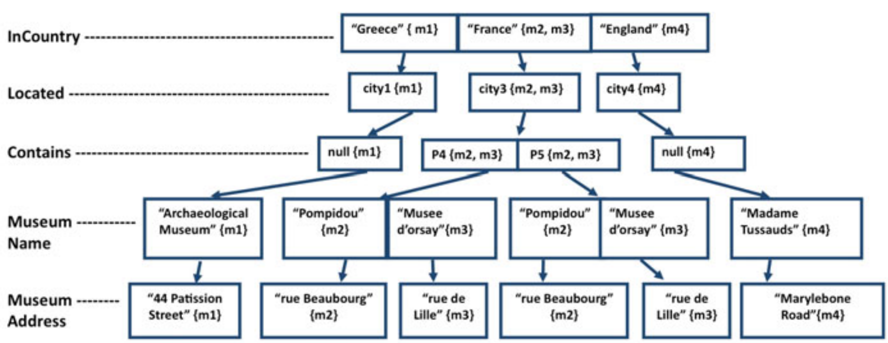


Fig. 2. Final prefix-tree for the museum class instances

Subsumption-driven Key Retrieval. For each set of RDF sources, the method $ClassKeyRetrieval$ applies a depth-first retrieval of the keys by exploiting the subsumption relation between classes declared in the ontology. $ClassKeyRetrieval$ method takes an instantiated class and a possible set of already known keys as input and calculates its complete set of keys. After creating the final prefix-tree of the considered class instances, the $UnKeyFinder$ method is called for retrieving the undetermined and the non keys. Then, the method is recursively called for the set of subclasses using the updated known key set.

ClassKeyRetrieval
Input: C: class; $KnownKeys$:=set of known keys
Output: $CKeys$: the complete set of keys of the class C.
if $class$ has declared properties **then**
  $tripleList$.add(all triples of C)
  if $tripleList$ is not empty **then**
    $rootNode$:= Create-intermediate-prefix-tree(tripleList, C)
    $newRootNode$:=Create-final-prefix-tree($rootNode$)
    $propNo := 0$
    $UNKeySet$:= UnKeyFinder($newRootNode$,$propNo$, $KnownKeys$)
    $keys := ExtractKeysFromUNKeySet(UNKeySet, C)$
    $CKeys := refine(KnownKeys.add(keys))$
  end if
end if
for all subClass C_i of C **do**
  ClassKeyRetrieval(C_i, $CKeys$)
end for
return $CKeys$

UNK-Finder: Undermined Key and Non Key Finder. The UNKey is the set of undetermined keys and non-keys. The process of the algorithm begins from the root of the prefix tree and makes a depth-first traversal of it. The input of the algorithm is the current root of the tree, its attribute number and the known keys. This method searches the longest path p from the root to a node having a URI list containing more than one URI. p represents the maximal set of properties expressing either a non-key or an undermined key.

To ensure the scalability of the key discovery, KD2R performs two kinds of pruning: (i) the subsumption relation between classes is exploited to prune the key discovery thanks to the set of inherited keys, (ii) the anti-monotonic characteristic of the non-keys and undermined keys is also used to avoid computing the redundant non keys and undermined keys, i.e. if {ABC} is a non key (resp. an undermined key) then all the subsets of {ABC} are also non keys (resp. undetermined keys) and (iii) the monotonic characteristic of keys is used to avoid exploring the descendants of a node representing only one instance.

UNK-Finder
Input: $root$: node of the prefix tree; $propNo$: attribute Number; $knownKeys$: given keys.
Output: $UNKeySet$: the set of discovered undermined keys and non-keys.
add $propNo$ to the $curUNKey$
if $root$ is a leaf **then**
  for all $cells$ in the $root$ **do**
    if $cell.PropertyList > 1$ **then**
      add $curUNKey$ to the $UNKeySet$
      break
    end if
  end for
  remove $propNo$ from $curUNKey$
  if $root$ has more that one cell AND at least one of the cells has $PropertyList > 1$ **then**

```
      add curUNKey to the UNKeySet
    end if
  else
    if there is only one URI then
      return
    end if
    for all cells in the root do
      if curUNKey is not contained in knownKeys Set then
        UNK-Finder(cell.getChild,propNo+1)
      end if
    end for
    remove propNo from curUNKey
    if there is more that one cell in the root then
      if curUNKey is already contained in the UNKeySet then
        return
      end if
      childNodeList := all the children of the cells in the root node
      mergedTree := mergeNodes(childNodeList)
      if curUNKey is not contained in knownKeys Set then
        UNK-Finder(mergedTree,propNo+1)
      end if
    end if
  end if
  return UNKeySet
```

Example of UNK-Finder. We illustrate the UNK-Finder algorithm on the final prefix-tree shown in figure 2. The method begins with the first node and more specifically with the cell containing the value " *Greece*". The property number of the cell, 0, is added on to $curUNKey$. Since the $URIList$ of this cell has size one -thanks to the pruning step- the algorithm will not examine its children. Now the property number is removed. The algorithm moves to the next cell of the root node. The property number 0 is added in the $curUNKey$. This cell contains a $URIList$ with two elements in it. So recursively, we go to node with cell "*city3*". Now the $curUNKey$ is $(0, 1)$. We call the UNK-Finder for the child node of the "*city3*". Now the root node is the node with paintings $P4$ and $P5$ and the property number of the node is added to the $curUNKey$ $(0, 1, 2)$. The process continues with the child node of cell P4. In the $curUNKey$, attribute number 3 is added. Since the cell "Pompidou" has $URIList$ of size one the UNK-Finder will not continue with the child node of "*Pompidou*". The method will continue with the second cell of the root node which is "Musee d'Orsay". Like "*Pompidou*", and since "*Musee d'Orsay*" has $URIList$ size one, the UNK-Finder will not be called for its child node. The UNK-Finder has been called for each cell of the node. The property number of the node is removed from the $curUNKey$ which now is $(0, 1, 2)$. In this step the child nodes of this node are merged and the UNK-Finder is applied to the mergedTree. The UNK-Finder is executed for the new merged node which consists of two cells, "*rue Beaubourg*" and "*rue de Lille*". The attribute number of the merged node is added in the $curUNKey$ which is $(0, 1, 2, 4)$. Since we are in the leaf and there is a cell in the node with $URIList$ bigger than one, the $curUNKey$ (0,1,2) is added in the $UNKeySet$. This means that there are more than two instances that have exactly the same values for the properties in the $curUNKey$. The process of discovering the $UNKeySet$ continues in the same way. In this specific example the final $UNKey$ set consists of one composite $UNKey$, which is {contains, located, inCountry}.

Extraction of keys from UNKeys. In order to compute the set of minimal keys from the $UNKey$ set, we first compute for each UNKey the complement set. Then we apply the Cartesian product on the obtained complement sets. Finally, we remove the non-minimal keys from the obtained multi-set of keys.

In the museum example we have only one $UNKey$ which is {contains, located, inCountry}. The complement set of this $UNKey$ is {{MuseumAddress}, {MuseumName}}. The process finishes by adding the two keys to the $KeySet$. The keys in the $KeySet$ are $MuseumAddress$ and $MuseumName$.

4 First Experiments

We have implemented and tested our method on two datasets that have been used in the Ontology Alignment Evaluation Intitiative (OAEI, `http://oaei.ontologymatching.org/2010/`). UNA is declared for each RDF file of the two datasets. Since the two ontologies have been enriched by expert keys, we have compared our results to the set of these existing keys.

Restaurant dataset. The first dataset $D1$ describes 1729 instances (classes $Restaurant$ and $Address$). In the provided Ontology, Restaurants are described using the following properties: $name$, $phoneNumber$, $hasCategory$, $hasAddress$. Addresses are described using: $street$, $city$, $Inverse(hasAddress)$. The first RDF file f1 describes 113 $Address$ instances and 113 $Restaurant$ instances. The second RDF file f2 describes 641 $Restaurant$ instances and 752 $Address$ instances.

Person dataset. The second dataset D2 consists of 3200 instances of $Person$ and $Address$. In the ontology, a person is described by the following properties: $givenName$, $state$, $surname$, $dateOfBirth$, $socSecurityId$, $phoneNumber$, age and finally $hasAddress$. An Address is described by the properties: $street$, $houseNumber$, $postcode$, $isInSuburb$ and finally $inverse(hasAddress)$. The first and the second RDF files contain each of them 500 instances of the class $Person$ and 500 instances of $Address$. The third file contains 600 $Person$ instances and 600 $Address$ instances.

To examine the results of our method we compared the KD2R keys with the keys given by an expert. 10% of found keys are equal to the expert keys and 10% are bigger (i.e., contain more properties). The first case is the best we can come up with since our results agree with the expert ones. The second case arises when an expert makes a mistake and declares as keys properties that are not in fact real keys. This means that we detect erroneous keys given by an expert. For instance, the expert has declared that phoneNumber is a key. We are sure that the expert has made a mistake since in our data we can find two different restaurants with the same phone number (managed by the same organization). These two cases (20% of our found keys) represent the definite minimal keys that we extract using the given datasets. Another 20% of KD2R keys are keys that are smaller compared to the expert keys. It is possible to face this case when the given data are not sufficient to find more specific keys. Finally the 60% of the found keys are keys that are not declared by the expert. For example we find that $Inverse(hasAddress)$ can be a key for the address, a property that the expert did not take into account and seems to be relevant (a museum has only one address).

Thus, KD2R may find keys that are not specific enough (the more the data are numerous the more the discovered keys are accurate). However, this method can also find keys that are equal to the expert ones or keys which are missed by the expert.

5 Conclusions and Future work

In this paper, we have described the method KD2R which aims to discover keys in RDF data in order to use them in a reconciliation method. These data conform to the same ontology and are described in RDF files for which the UNA is fulfilled. KD2R takes into account the properties that the RDF files may have : incompleteness and multi-valuation. Since the data may be numerous, the method discovers maximal undetermined/non keys that are used to compute keys and merge them if keys are discovered using different datasets. Furthermore, the approach exploits key inheritance due to subsumption relations between classes to prune the key search for a given class. The first experiments have been conducted on two datasets exploited in the OAEI evaluation initiative. We have compared the retrieved keys with keys given by an expert. Some of the found keys are less specific than the expert ones but errors of the expert can also be detected. We plan to test our approach on more heterogeneous data and extend our method in order to be able to work even when the UNA is not fulfilled.

References

1. Bhattacharya, I., Getoor, L.: Collective entity resolution in relational data. ACM Trans. Knowl. Discov. Data 1 (March 2007)
2. Dong, X., Halevy, A., Madhavan, J.: Reference reconciliation in complex information spaces. In: Proceedings of the 2005 ACM, SIGMOD 2005, NY, USA, pp. 85–96 (2005)
3. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering 19, 1–16 (2007)
4. Huhtala, Y., Kärkkäinen, J., Porkka, P., Toivonen, H.: Tane: An efficient algorithm for discovering functional and approximate dependencies. Comput. J. 42(2), 100–111 (1999)
5. Low, W.L., Lee, M.L., Ling, T.W.: A knowledge-based approach for duplicate elimination in data cleaning. Information Systemes 26, 585–606 (2001)
6. Nikolov, A., Motta, E.: Data linking: Capturing and utilising implicit schema-level relations. In: Proceedings of Linked Data on the Web Workshop Collocated with WWW 2010 (2010)
7. Saïs, F., Pernelle, N., Rousset, M.-C.: Combining a logical and a numerical method for data reconciliation. Journal on Data Semantics 12, 66–94 (2009)
8. Sismanis, Y., Brown, P., Haas, P.J., Reinwald, B.: Gordian: efficient and scalable discovery of composite keys. In: Proceedings of the 32nd International Conference VLDB 2006, pp. 691–702. VLDB Endowment (2006)
9. Wang, D.Z., Dong, X.L., Sarma, A.D., Franklin, M.J., Halevy, A.Y.: Functional dependency generation and applications in pay-as-you-go data integration systems. In: 12th International Workshop on the Web and Databases (2009)
10. Winkler, W.E.: Overview of record linkage and current research directions. Tech. rep., Bureau of the Census (2006)

A Framework for Change Impact Analysis of Ontology-Driven Content-Based Systems

Yalemisew M. Abgaz, Muhammad Javed, and Claus Pahl

Centre for Next Generation Localization (CNGL),
School of Computing, Dublin City University, Dublin 9, Ireland
{yabgaz,mjaved,cpahl}@computing.dcu.ie

Abstract. The trend in content-based systems (CBSs) is shifting towards the use of ontologies to semantically enrich the content and increase its accessibility. The growing need of semantically rich content becomes a driving force for building ontology-driven content-based systems (ODCBSs). The building blocks of ODCBSs are ontologies, content and annotations, forming a layered information model. In most ODCBSs, changes in the content, in the ontology or in the annotation are inevitable and are observed on a daily basis. Any change on one layer of the architecture has an impact within the layer and on the other layers. Impact analysis in large multi-ontology CBS is a manual, time consuming and labour intensive process. It is done only when it is necessary. Based on observation and empirical analysis, we propose a conceptual framework for dependency-based impact analysis and identify the possible impacts and their causes, the dependency among entities, their severity and factors affecting impact analysis process in ODCBSs.

Keywords: Ontology evolution, Change impact analysis, Content-based systems, Ontology-driven content-based systems.

1 Introduction

Ontologies became ubiquitous and standard means of embedding semantic information in most of the existing content-based applications [1]. In such applications, ontologies are used to semantically enrich content and services. Many applications are integrated with ontologies using semantic annotation to identify information, process them and reason about subjects of interest. Content-based systems (CBSs) become dependent on ontologies to provide a better service for developers, designers and end-users of such systems. This is achieved by using ontologies to annotate the target content so that both human and computer systems can understand what meaning is exactly conveyed in it [2]. This process leads to the emergence of ontology-driven content-based systems (ODCBSs).

Despite the promising benefits, ODCBSs face challenges. One of the major challenges is the changing nature of content and thus the dynamic evolution of the ontologies that support the ODCBS [3][4]. The interdependence between the content and the ontologies further aggravates the challenge in that, a change in one layer affects entities in the given layer and in all dependent layers. For relatively large ODCBSs, determining the impacts of a single change operation

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 402–411, 2011.
© Springer-Verlag Berlin Heidelberg 2011

is difficult, time consuming and often doesn't guarantee a complete solution. To solve this problem, we propose a conceptual framework for dependency-based analysis of impacts of changes in ODCBSs to identify impacts, affected entities and to determine the severity of the impacts. The framework is used to provide terminological and formal guidance for analytical and operational change support. In this context the determination of change impact is a crucial first activity. We used graphs for the formalization of the ODCBS layers to facilitate the dependency analysis and impact determination process.

The term *impact* refers to the effect of change of entities due to the application of a change operation on one or more of the entities in the ODCBS [5][3][4][6]. By *impact analysis* we mean the process of identifying and determining the impacts of a requested change operation on the ODCBSs layers.

Impact analysis identifies the impacts of a change operation before it is permanently implemented. Due to frequent changes in the content and continuing evolution of the ontologies, impact analysis becomes an important step in the evolution of ODCBSs. The core contribution of this paper is a conceptual framework for dependency-based impact analysis using empirical identification of:

- the possible impacts and their categorization.
- the causes of impacts in the content, ontology and annotation layers.
- the dependencies and the types of dependencies that exist between a changing entity and other entities.
- the severity of each of the impacts on the ODCBS and dependent systems.

For a given change request, the knowledge of the above discovered inputs ensures earlier visibility of impacts and smooth evolution by automatically identifying the affected entities and impacts. It guarantees accurate execution of nothing but the desired changes with minimum impacts and it reduces risk on dependent systems by taking prior preventive measures to reduce the impacts.

This paper is organized as follows: Section 2 describes the layers in ODCBSs and Section 3 focuses on graph-based representation of each layer of the ODCBS. Section 4 presents dependencies in ODCBS and section 5 focuses on impacts of changes. Discussion and related work are given in section 6 and conclusion and future work in section 7.

2 Ontology-Driven Content-Based Systems

ODCBSs are systems that use ontologies to semantically enrich the content they provide. The aim of ODCBSs is to facilitate accessibility of content for both humans and machines by integrating semantics in the content using ontologies.

2.1 Layered Architecture of ODCBSs

The ODCBSs is composed of three different layers. The first layer is the ontology layer (represented using OWL), the second is the annotation layer (represented using RDF triples) and the third one is the content layer (set of documents). The layered architecture is presented in (Fig. 1).

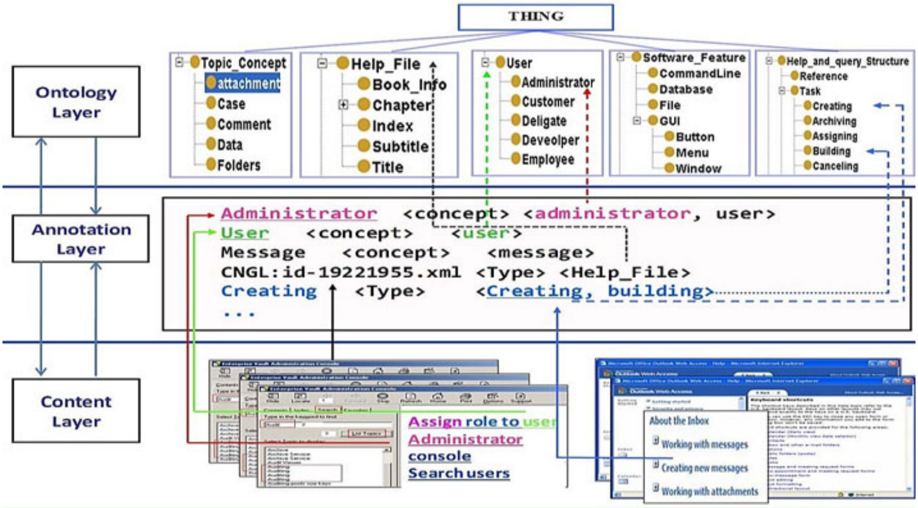


Fig. 1. Layered architecture of ODCBSs

Ontology Layer. Ontology is a specification of a shared conceptualization of a domain [7]. This means ontologies provide a common ground for understanding, conceptualization, representation and interpretation of domain concepts uniformly across different systems, languages and formats. They provide a representation of knowledge that allows machines to reason about known facts and generate new knowledge from them.

In our ODCBSs architecture, ontologies become crucial component as many CBSs are integrating ontologies for semantic annotation. A growing number of applications use ontologies to the extent that makes ontologies unavoidable integral parts of the applications.

The ontology layer is subject to change due to a change in specification, representation or conceptualization of knowledge [8]. New concepts are added, existing ones deleted or modified. In frequently evolving domains these changes are numerous and have impact on dependent entities in the ODCBSs.

Content Layer. Content, in this paper, refers to any digital information that is in a textual format that contains structured or semi structured documents, web pages, executable content, software help files etc [9][10]. ODCBSs essentially deal with content in a form of books, web pages, blogs, news papers, software products, documentations, help files reports, publications etc [9].

Content in ODCBSs is a collection of content documents that change frequently. This means new content documents are produced, existing ones are modified, edited or deleted frequently to provide up-to-date information. Such activities have impacts on dependent entities in the overall ODCBS.

Annotation Layer. Annotation is a process of linking content with ontology entities to provide better semantics to the content. The aim of semantic annotation is to explicitly identify concepts and relationships between concepts in the content [1]. In any application that makes use of ontologies, the target content which needs to be semantically enriched is required to have an explicit link, at least to one or more elements in the ontology.

In our ODCBS, the annotation is treated as a separate layer to allow independence of the annotation data from the content, to achieve visibility and better impact analysis. The annotation layer is one of the most interactive and frequently changing layers. There are a number of triples added, modified or deleted in this layer. This layer is highly dependent on both the content and the ontology layer. Any change in the other two layers affect the annotation layer which carries all the semantics related to the content.

In the annotation layer, a document or part of a document is treated as instances of one or more concepts. For example $< CNGL : id{-}19221955.xml, rdf : type, CNGL : Help\_File >$ indicates that "$CNGL : id - 19221955.xml$" is an instance of the concept "$CNGL : Help\_File$" (Fig. 2).

2.2 Running Example

We conducted empirical analysis on database systems, univeristy administration [8] and software help management systems domains [10]. Software help management systems domain is selected to serve as a running example (Fig. 2). Suppose we want to find all the impacts of **Delete Class(*Activity*)** operation. The requested operation is deletion and the target entity is concept *Activity*. To identify the dependent entities, we need to know if the change is applied in a cascaded strategy or not [3][11]. If we choose cascade strategy, meaning if the deletion of the concept *Activity*, deletes all its subclasses, we should identify all the subclasses of *Activity* and their subclasses iteratively, which are { *Archiving, ArchivingEmail, Deleting, DeletingDirectory...* } and save them in a list of dependent entities. We further identify all the axioms {(*Archiving*, subclassOf, *Activity*), (*Deleting*, subclassOf, *Activity*)...} , instances {*CNGL:id-19221955.xml...*} and so on.

We identify what kinds of changes are required to each of these dependent entities to make the original change request effective. In the case of cascade delete strategy, we have a set of cascaded change operations like {Delete Concept (*Adding*)..., Delete Instance (*CNGL:id-19221955.xml*)... Delete Axiom (*Archiving*, subclassOf, *Activity*) ... Delete Class (*Activity*)}. The set of change operations on the entities imply their effects. The impacts of these changes, for example, are the removal of the target entities (section 5.1). Once we get the impact set, we attach a severity value to each impact (section 5.2).

In general, the impact varies following the type and the taxonomic position of the target entity, the type of operation and the change strategy implemented. For example, the deletion of the concept *Activity* caused many cascaded operations, due to its structure and the change strategy. However, if the concept *Activity* doesn't have dependent entities or if the change strategy is different, the final change operations will be different and so is the impact set.

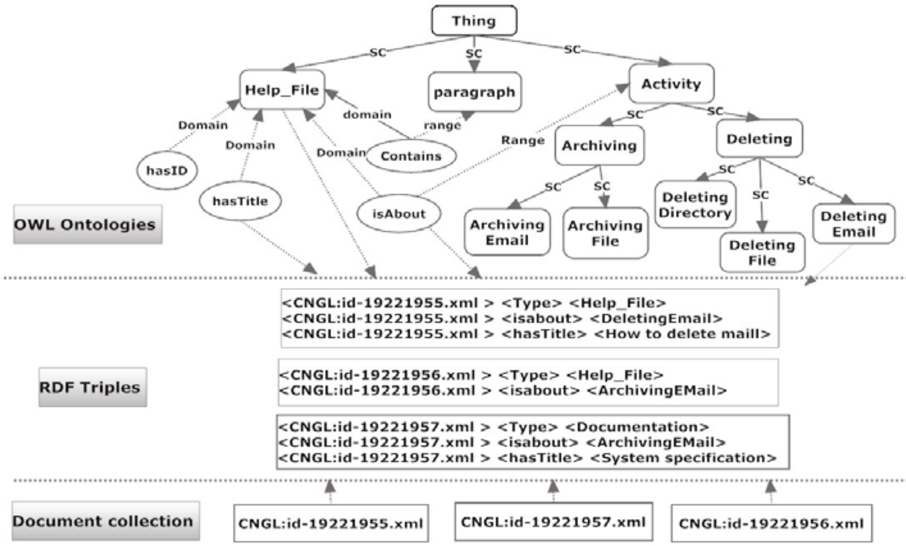


Fig. 2. An example of ODCBS for software help systems

3 Graph-Based Representation of ODCBS

The ODCBS can be represented using graph-based formalism. Graphs are selected for their known efficiency and similarity to ontology taxonomy. In our ODCBS, the ontology and the annotation are represented as graphs and the content is represented as a set of documents. The document set serves as a node (of type instances) in the annotation layer.

An ODCBS is represented as graph $G = \{G_o\} \cup \{G_a\} \cup \{Cont\}$ where: G_o is the ontology graph, G_a is the annotation graph and $Cont$ is the content set.

An ontology O is represented by a direct labelled graph $G_o = (N_o, E_o)$ where: $N_o = \{n_{o1}, n_{o2}, \ldots, n_{om}\}$ is a finite set of labelled nodes that represent classes, data properties, object properties etc. $E_o = \{e_{o1}, e_{o2} \ldots, e_{om}\}$ is a finite set of labelled edges and $e_{oi} = (n_1, \alpha, n_2)$ where: n_1 and n_2 are members of N_o and the label of an edge represented by $\alpha = \{$subclassOf, intersectionOf, minCardinality, maxCardinality...$\}$. The labels may indicate the relationship (dependency) between the nodes.

A content represented by $Cont$ can be viewed as a set of documents $D = \{d_1, d_2, d_3 .... d_n\}$ where: d_i represents a single document or part of a document which can be mapped to nodes in the annotation graph.

An annotation $Anot$ is represented by a direct labelled graph $G_a = (N_a, E_a)$ where: N_a and E_a are finite set of labelled nodes and edges respectively. An edge $E_a = (n_{a1}, \alpha_a, n_{a2})$ where $n_{a1} \in \{Cont\}$ as a subject, $n_{a2} \in \{Cont\} \cup \{O\}$ as an object and $\alpha_a \in \{O\}$ as a predicate. The nodes are mapped to a non-empty string.

The type of any node is given by a function $type(n)$ that maps the node to its type (class, instance, data property, object property...). The label of any edge $e = (n_1, \alpha, n_2)$, which is α , is a string given by a function $label(e)$. All the edges of a node n are given by a function $edges(n)$. It returns all the edges as (n, α, m) where n is the target node and m is any node linked to n via α.

4 Dependency in ODCBSs

Dependency analysis is a process of identifying the artefacts that are dependent on a given entity in an ontology, content or annotation. Dependency analysis identifies all entities that depend on a target entity. Identifying these dependencies and their types has significant contribution to impact analysis process.

4.1 Types of Dependencies

Using the empirical study, we identified the following dependency types that play a major role in the impact analysis process in ODCBSs. We also observed that there is no sharp demarcation between the identified dependency types, thus, they are not mutually exclusive.

Structural Dependency/Semantic Dependency. Structural dependency refers to the syntactic dependency between two nodes. When a node changes, it will have a structural impact on adjacent nodes. Semantic dependency refers to the semantic relation that exists between two nodes. A change in one node e.g. *Activity*, causes a change in the semantic meaning or the interpretation of the dependent nodes *(Archiving and Deleting)*.

Direct Dependency/Indirect Dependency. Direct dependency is the dependency that exist between two adjacent nodes(n_1, n_2). This means, there is an edge $e_i = (n_1, \alpha, n_2)$. Indirect dependency is a dependency of a node on another by a transitive or intermediate relationship. There exist a set of intermediate edges $(n_1, \alpha, n_x)(n_x, \alpha, n_y)...(n_z, \alpha, n_2)$ that link the two nodes. For example, in Fig. 2 there is a direct dependency between *Activity* and *Deleting* and indirect dependency between *Activity* and *Deleting_File*.

Total Dependency/Partial Dependency. Total dependency refers a dependency when a target node depends only on a single node (articulation node). Partial dependency refers to a dependency when the existence of a node depends on more than one node.

4.2 Dependency within and among Layers

Dependency in the Ontology Layer. A change of an entity in one ontology first affects the dependent entities within the ontology. Identifying the dependencies in this layer is a crucial step. These dependencies are identified based on the inheritance (such as *is-a* relationships) association (such as *has* relationships) and so on. There is also dependency across ontologies. We present one of the empirically identified dependencies from our case study.

Concept-Concept Dependency: Given two class nodes c_i and c_j in G_o, c_i is dependent on c_j represented by $dep(c_i, c_j)$, if there exist an edge $e_i = (n_1, \alpha, n_2) \in G_o$ such that $(n_1 = c_i) \wedge (n_2 = c_j) \wedge (label(e_i) = \text{“}SubclassOf\text{”}) \wedge (type(n_1) = type(n_2) = \text{“}Class\text{”})$. Concept-concept dependency is transitive.

Dependency in the Annotation Layer. In this layer we have two directions of dependency. The first refers to the dependency of the annotation on the content layer (Content-Annotation Dependency). An annotation a_i in the annotation layer is dependent on d_i in the content layer, represented by $dep(a_i, d_i)$, if there exist an edge $e_a = \{n_{ai}, \alpha_a, n_{aj}\} \in G_a$ such that $(n_{ai} = d_i) \vee (n_{aj} = d_i)$. This means a_i is dependent on document d_i if the document is used as a subject or an object of the annotation triple.

The second refers to the dependency of the annotation on the ontology (Ontology-Annotation Dependency). An annotation a_i in the annotation layer is dependent on o_i in the ontology layer, represented by $dep(a_i, o_i)$, if there exist an edge $e_a = \{n_{ai}, \alpha_a, n_{aj}\} \in G_a$ such that $(\alpha_a = o_i) \vee (n_{aj} = o_i)$.

Dependency in the Content Layer. Intra content dependency (Content-Subcontent Dependency) is a dependency that exists between a document and its subsections. This includes the dependency of section, title, paragraph, step, procedure etc on a containing document. Whenever the content in the documents are updated, for example, deletion of a section, addition of steps etc, affect all the section and documents related to the document.

5 Impact of Changes in ODCBSs

Changes in ODCBS have diverse impacts on the individual entities of the layers and on the overall ODCBS. Impact analysis identifies the possible impacts of proposed changes and determine the severity of the impacts [12][6][13]. Determining the impacts that exist in the ODCBS and deciding the severity of the impacts are essential steps for impacts analysis.

5.1 Types of Ontology Change Impact

In ODCBS we can categorize impacts using different criteria. The categorization paves a way to better understand impacts in ODCBS and makes the analysis and the determination process understandable and suitable for implementation.

Structural and Semantic Impact. Structural impact is an impact that changes the structural relationship between the elements of the ODCBS. Structural changes are the main reasons for structural impacts. Structural changes include any atomic or composite changes[3] that are applied on concepts, properties, axioms and restrictions. Structural impact occurs when we request a change that affects the taxonomy of the existing ODCBS. Semantic impact occurs due to a change in the interpretation of entities due to structural changes. Structural and semantic changes are discussed in [5] and the impacts are discussed in [10].

Addition and Deletion Impact. The categorization of impacts of addition and deletion became visible in the empirical study. In ODCBSs, the impacts of addition operation are different from the impacts of deletion operation. Furthermore, the complexity of the impacts differ one another. In such situation it is intuitive to treat the operations and their impacts separately.

Ontology, Annotation and Content Impact. Impacts can further be divided based on the target layer. This categorization allows us to know which layers are affected by the change operation. Impacts in the ontology layer include all impacts on the entities defined in the given ontology. Such impacts need careful treatment as they further affect the annotation layer. Impacts in the annotation layer primarily revolve around the triples. However, a triple contains the subject (usually a reference to the content), the predicate (usually a property in the otology) and the object (usually a reference to the content or the ontology). Thus, impact analysis in the annotation layer makes use of these three elements and tries to find what impacts the change operation will have on them. Impacts in the content layer concentrate around the documents. The addition, deletion or the modification of the content or part of the content is treated as an impact and affects the other two layers.

$\mathcal{A}$Box and $\mathcal{T}$Box Impacts. Impacts of a change operation can be viewed from the perspective of the kind of statement it affects. Change operations may have an impact on the $\mathcal{A}$Box or $\mathcal{T}$Box statements. $\mathcal{T}$Box statements are affected by operations that change the concepts and axioms related to the terminology in the ontology. The impact of such change operation concentrates around the satisfiability of the terminologies in the $\mathcal{T}$Box and identifying them helps us to pinpoint the causes of contradiction. $\mathcal{A}$Box statements are affected by operations that change the axioms related to annotation instances (individuals) in the assertion box. The impact of operations on the $\mathcal{A}$Box axioms may result invalidity (unable to interpret a give instance with respect to a given ontology) [5].

Impact analysis is mainly affected by the change strategy implemented at the time of evolution. The content engineer may choose to delete all orphaned entities or link them to their parents or to the root class. The different types of dependencies are crucial at this stage. For example, direct dependency is used when attach to root strategy is used. Total dependency is used when cascade delete strategy is used. Partially dependent entities remain intact in the system.

5.2 Severity of Change Impacts

Severity is defined as the extent of impact of a change operation in the ODCBS. The impact is measured qualitatively using consistency and validity, or quantitatively using number of change operations required, ontology elements affected and cascaded effect on dependent entities.

The impact can be on the structure or on the semantic, satisfiability or validity of the existing ontology. We analyzed how different change operations impact the ODCBS. This gives us a better understanding of which operations under what condition have a more severe impact. We used Operation Severity Function (OSF) that maps operations to severity values on a scale of 0 to 100 based on user defined configuration. To indicate the severity as a qualitative scale, we categorized them into four scales: less impact (0-25), medium impact (26-50), high impact (51-75) and crucial impact (76-100) [10]. Based on this scale the user can determine the severity of an impact relative to his ODCBS.

6 Discussion and Related Work

Using a software help management system [10], we tested our approach empirically. From the case study, we found out that the proposed solution is useful and feasible to analyze impacts of changes in ODCBSs. To our knowledge, there are few research conducted to analyze the impacts of changes in ODCBSs. But there are significant research in the area of software change impact analysis in general [14] [6] [15].

The author in [14] conducted change impact analysis on commercial-Off-The-Shelf software. They identified different reasons for software change and classify software impacts as direct or indirect, and structural or semantic impacts. They further conducted structural analysis and semantic analysis using reachability graphs by implementing transitive closure algorithms. They focus on the syntactic relationship between software modules where as we focus on structural and semantic changes with detailed semantics. In [6] the authors presented a knowledge-based system for change impact analysis on software architecture. They proposed an architectural software component model on which they defined change propagation process and used graphs to capture architecture elements and their relationships. The authors conducted impact analysis using rules that define change propagation. Their work is similar to ours but with a significant difference in the domain and in the impact determination approach.

The work in [5] discusses consistent evolution of OWL ontologies with the aim of guaranteeing consistency. Their work identifies structural, logical and user-defined consistency, but focuses on determination of validity of instances, whereas our work focuses on the overall impact analysis of change operations.

7 Conclusion and Future Work

Based on our empirical analysis, we identified different dependencies that exist within and among ODCBS layers. We identified dependencies and impacts in a conceptual framework then further categorized them based on different criteria that will serve as an input for the impact analysis process. We also investigated the severity of the impacts.

The contribution of our work is the empirical observation of dependencies in ODCBS, types of impacts and factors affecting impact analysis. This will enable us to ensure earlier visibility of impacts of changes prior to their implementation, automatic capturing and presentation, accurate determination and reducing their impacts on dependent systems. Our next step will further extend the work to optimize the implementation of change operations to ensure minimum and less severe impact.

Acknowledgment. This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/CE/I1142 as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University (DCU).

References

1. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management:requirements and survey of the state of the art. Web Semantics: Science, Services and Agents on World Wide Web 4 (1), 14–28 (2006)
2. Reeve, L., Han, H.: Survey of semantic annotation platforms. In: SAC 2005: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 1634–1638 (2005)
3. Stojanovic, L.: Methods and tools for ontology evolution. PhD thesis, University of Karlsruhe (2004)
4. Plessers, P., De Troyer, O., Casteleyn, S.: Understanding ontology evolution: A change detection approach. Web Semantics: Science, Services and Agents on the World Wide Web 5(1), 39–49 (2007)
5. Qin, L., Atluri, V.: Evaluating the validity of data instances against ontology evolution over the semantic web. Information and Software Technology 51(1), 83–97 (2009)
6. Hassan, M.O., Deruelle, L., Basson, H.: A knowledge-based system for change impact analysis on software architecture. In: Fourth International Conference on Research Challenges in Information Science (RCIS), pp. 545–556 (May 2010)
7. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)
8. Javed, M., Abgaz, Y., Pahl, C.: A pattern-based framework of change operators for ontology evolution. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 544–553. Springer, Heidelberg (2009)
9. Gruhn, V., Pahl, C., Wever, M.: Data Model Evolution as Basis of Business Process Management. In: Papazoglou, M.P. (ed.) ER 1995 and OOER 1995. LNCS, vol. 1021, pp. 270–281. Springer, Heidelberg (1995)
10. Abgaz, Y., Javed, M., Pahl, C.: Empirical Analysis of Impacts of Instance-driven Changes in Ontologies. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 368–377. Springer, Heidelberg (2010)
11. Haase, P., Stojanovic, L.: Consistent evolution of owl ontologies. In: Proceedings of the Second European Semantic Web Conference, Heraklion, Greece (2005)
12. Lee, M., Offutt, A., Alexander, R.: Algorithmic analysis of the impacts of changes to object-oriented software. In: 34th International Conference on Technology of Object-Oriented Languages and Systems, TOOLS 34, pp. 61–70 (2000)
13. Arnold, R.S., Bohner, S.A.: Impact analysis - towards a framework for comparison. In: Proceedings of the Conference on Software Maintenance. ICSM 1993, pp. 292–301. IEEE Computer Society, Washington, DC (1993)
14. Bohner, S.A.: Extending software change impact analysis into cots components. In: Proceedings of 27th Annual NASA Goddard/IEEE Software Engineering Workshop, pp. 175–182 (December 2002)
15. Sherriff, M., Williams, L.: Empirical software change impact analysis using singular value decomposition. In: Proceedings of the 2008 International Conference on Software Testing, Verification, and Validation, pp. 268–277. IEEE Computer Society, Washington, DC (2008)

Introducing Layers of Abstraction to Semantic Web Programming*

Bernhard G. Humm and Alexey Korobov

Hochschule Darmstadt – University of Applied Sciences,
Haardtring 100, 64295 Darmstadt, Germany
Bernhard.Humm@h-da.de

Abstract. Developers of ontologies and Semantic Web applications have to decide on languages and environments for developing the ontology schema, asserting statements, specifying and executing queries, specifying rules, and inferencing. Such languages and environments are not well-integrated and lack common abstraction mechanisms. This paper presents a concept framework to alleviate those problems. This is demonstrated by a complex sample application: reasoning over business process models.

Keywords: Semantic Web, software engineering, business process models, Lisp, Prolog.

1 Introduction

Developers of ontologies and Semantic Web applications often face the following questions:

- Which language and environment shall be used for *developing the ontology schema*? For example, when developing a Semantic Web application for reasoning over business process models, the schema may include classes like "BusinessProcess" and "Activity" and properties like "hasBusinessProcess". In many projects, a graphical ontology development environment like Protégé or TopBraid Composer is chosen to develop the ontology schema in languages such as RDF, RDFS, and OWL.
- Which language and environment shall be used for *asserting statements*? Example in the business process domain: "CompanyABC has BusinessProcess TravelManagement". While such statements may be added manually to an ontology in a graphical ontology environment, they usually have to be asserted dynamically by an application. For this, Semantic Web frameworks are used which allow to embed Semantic Web languages such as RDF, RDFS, and OWL into general purpose programming languages like Java. Examples for mainstream Semantic Web frameworks are Jena and Sesame.

---

* This research was funded by Zentrum für Forschung und Entwicklung (ZFE), Hochschule Darmstadt – University of Applied Sciences under grant number 419 327 01.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 412–423, 2011.
© Springer-Verlag Berlin Heidelberg 2011

- Which language and environment shall be used for *specifying and executing queries*? Example: "Which companies have business processes for travel management?" Semantic Web query languages like SPARQL allow for querying ontologies. Where SPARQL queries usually can be executed from graphical ontology development environments, this usually is used for demonstration and testing purposes only. As with asserting statements, queries usually have to be executed dynamically from a Semantic Web application and the query results are processed further, e.g. displayed on a Web page. Again, Semantic Web frameworks allow to embed Semantic Web query languages such a SPARQL and further process the results.
- Which language and environment shall be used for *specifying rules and inferencing*? Example rule: "If two business processes have similar names then they are likely to be in the same business domain". Standardized Semantic Web rule languages like RIF or proprietary rules languages like Sesame Rules may be embedded in Semantic Web frameworks.

These questions lead to a number of issues or problems.

- *Developing ontology schema*: Manually developing an ontology schema in RDF/XML is not feasible due to its most verbose syntax. N3 is the most concise and suitable syntax for developing ontology schemas manually. However, it still lacks abstraction mechanisms. Every statement must be developed in the simple triple notation. The only grouping mechanisms are semicolon and comma notations for avoiding repetitions of subjects and subject / predicate combinations. For example, it is not possible to define a concept "reification" with three input parameters (subject, predicate, object) and a blank node as an output parameter. Instead, each reification requires the (redundant) specification of three triples with predicates `rdf:subject`, `rdf:predicate`, and `rdf:object`.

  Graphical ontology development environments largely alleviate those problems, for instance, by providing wizards for reifying statements. However, developers cannot define similar abstractions on their own.
- *Asserting statements*: A single triple to be inserted into a RDF store via deserializing RDF / RDFS / OWL takes a single line of N3 code. In contrast, adding the same triple programmatically via a Semantic Web framework like Jena or Sesame takes about 15 lines of Java Code[1]. This includes repository and connection handling, instantiating objects for resources and literals, asserting statements, and exception handling. So, simple statements cannot be expressed in a concise way as is possible in N3.

  On the other hand, Java offers mechanisms for defining abstractions such as classes and methods. A method "reify" taking "subject", "predicate", and "object" as input parameters and returning a reified blank node may be implemented once and then used many times wherever reification is needed in this form.
- *Specifying and executing queries*: SPARQL is a concise query language, similar to N3. However, it, too, misses abstraction mechanisms. Where SPARQL V1.1 introduces subqueries, it still lacks named, parameterized queries that can be invoked as

---

[1] Basis: code samples from the Sesame User Guide
`http://www.openrdf.org/doc/sesame2/users/ch08.html`

subqueries. For example, a query for all companies that provide business process ?x cannot be defined once and then re-used in many queries. Copying and pasting similar SPARQL WHERE parts is, therefore, common practice, leading to redundant code which is difficult to maintain.

Also, executing SPARQL queries from within Semantic Web frameworks is cumbersome. For example, in Sesame it takes about 20 lines of Java code to evaluate a one-line SPARQL query. This includes connection handling, query preparation and evaluation, iterating result set, identifying individual results, and exception handling.

- *Specifying rules and inferencing:* While SPARQL is *the* standard Semantic Web query language, a de-facto Semantic Web rule standard has not yet emerged. Embedding rules in Semantic Web frameworks faces the same usability issues as asserting statements and executing queries.

In total, different languages and environments, all with their strengths and weaknesses, but not well integrated in a useable fashion impede developing ontologies and Semantic Web applications. In particular, abstraction mechanisms in Semantic Web languages and technologies are limited. We have developed a framework for concepts that addresses those issues – see the following section.

2 A Concept Framework for Semantic Web Programming

2.1 Environment

For the implementation of the concept framework, we have chosen AllegroGraph, a commercial Semantic Web framework by Franz Inc.. AllegroGraph is based on Lisp, in particular Allegro Common Lisp, a professional implementation of the ANSI Common Lisp standard. It supports RDF, RDFS, SPARQL, and the OWL subset RDFS-Plus.

AllegroProlog is used as reasoning and query language. AllegroProlog is a Prolog implementation by Franz Inc., fully integrated in Lisp. It allows Prolog programming in Lisp notation.

It shall be noted, however, that the concept framework described in this paper is not specific to Lisp, Prolog, AllegroGraph, or AllegroProlog.

2.2 Our Use of the Term "Concept"

Encyclopedia Britannica defines *concept* as: "an abstract or generic idea generalized from particular instances" [1]. This definition is valid for our purposes. Additionally, our notion of concept is always in the context of an application domain for which an ontology or a Semantic Web application is developed. For example, in the application domain of business process models, *UML activity* [2] is a concept.

2.3 A DSL for Specifying Concepts

We have developed a simple Domain-Specific Language (DSL) [3], called *concept DSL,* that allows for specifying concepts. The *basic features* of the concept DSL are as follows:

- `define-concept` specifies a new named concept with *0..n* concept parameters.
- `triple` allows for using all provided RDF, RDFS, and OWL constructs as well as self-defined classes, instances, and properties.
- `<concept>` allows for using all lower-level, more concrete concepts previously defined via `define-concept` by their names.

In summary, concepts form trees with triples as leaves and other concepts as inner nodes.

Advanced features of the concept DSL are as follows:

- `&optional` allows specifying optional concept parameters.
- `local` allows using local variables within concept specifications. Local variables are particularly useful for introducing blank nodes and for generating URIs.
- `cond` allows for specifying pre-conditions to be checked before asserting statements, reasoning, and querying.
- `<name space>:<concept name>` allows for using identical concept names in different name spaces for different application contexts. They support the development of large ontologies. Where used, name space identifiers precede a concept name, separated by a colon.

In summary, a concept specification in a BNF-like notation [4] is as follows:

```
(define-concept <concept> (<parameter>*
                           [&optional <parameter>*])
  [(local <variable> <expression>)]
  [(cond <expression>)]
  ([<namespace>:]<concept> <parameter>*)*
  (triple <subject> <predicate> <object>)*)
```

Example: concept of an RDFS instance

```
(define-concept instance (uri class &optional label
comment)
  (triple uri !rdf:type class)
  (triple uri !rdfs:label label)
  (triple uri !rdfs:comment comment))
```

The concept specification uses `triple` only. The exclamation mark (Wilbur reader macro) indicates a Semantic Web URI.

Example: concept of a node in a graph:

```
(define-concept node (uri node-type graph label
&optional comment)
  (cond (sub-class node-type !modl:node))
  (instance uri node-type label comment)
  (triple graph !modl:contains uri))
```

The higher-level, more specific concept `node` uses the lower-level, more general concepts `instance` and `sub-class`.

2.4 Framework Implementation

At compile time, the concept framework parses each concept specification and generates the following source code.

1. *Lisp function for asserting statements*: The function has the same name and parameters as the concept. Optional concept parameters are being handled using Common Lisp's `&optional` feature. Local concept variables are being handled via Lisp variables. The expression following `cond` are implemented as pre-conditions. Triples with respective subject, predicate, and object are asserted to the triple store using the AllegroGraph built-in function `add-triple`. For lower-level concepts being used, the respective Lisp function is invoked and the parameters are passed.
2. *Prolog predicates for reasoning and querying*: The predicates are named after the concept and contain all mandatory parameters. Since Prolog does not support optional parameters, optional concept parameters are being handled by generating multiple predicates with increasing numbers of parameters. Local concept variables are handled via Prolog variables. Expressions following `cond` are implemented as conjunctive goal terms. Different predicates allow for reasoning and querying with and without local variables. The built-in AllegroProlog predicate `q-` is used to prove against asserted triples in the triple store. For lower-level concepts being used, the respective Prolog predicate with its parameters is used as a conjunctive goal term.

Fig. 1 illustrates concept specification and generated Lisp function and Prolog predicates by the example of the concept node.

In the example, the Lisp function `node` is generated from the concept `node` and may be used as follows.

```
(node !trv:req-app !uml:activity !trv:uml "request
approval")
```

The example shows a statement about a node in an UML activity diagram.

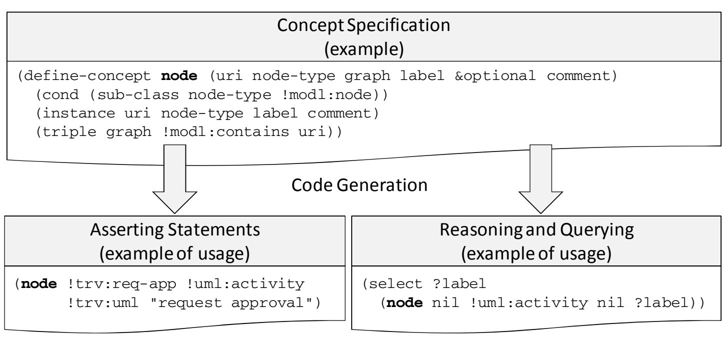


Fig. 1. Concept definition and code generation

A query using the generated Prolog predicate `node` may look like this:

```
(select ?label (node nil !uml:activity nil ?label))
```

The query returns the labels of all nodes of type `!uml:activity`, in this case `"request approval"`. `nil` indicates a don't care parameter value, e.g., the node's graph is irrelevant in this query.

The implementation of the concept framework is straight forward and comprises only about 100 lines of Lisp code excluding comments and blank lines. The core is the Lisp macro `define-concept` which generates Lisp code at compile time using the built-in Common Lisp macro processor.

3 Application: Reasoning over Business Process Models

We have applied the concept framework in a complex Semantic Web application that allows reasoning over business process models. In this section, we explain a sample application scenario, give an overview of the application, and show simple examples.

3.1 Application Scenario

Consider the following application scenario: Two companies decided to merge. To leverage synergies, their business processes shall be aligned. Business processes are modeled in numerous models in different formats in both companies, e.g., as UML activity diagrams, Event-Based Process Chain (EPC) diagrams, and Business Process Modeling Notation (BPMN) diagrams. The task of the application is to pre-select similar business process models to support human experts in their detailed analysis.

For this, we transform different business process models into an ontology and use reasoning mechanisms for detecting similarity between models.

3.2 Sample Business Processes

Consider, e.g., the process models for business travels in Fig. 2, one represented as an EPC diagram and the other one as a UML activity diagram. Both diagrams represent business processes for business travels – similar in content, but different in the modeling notations used as well as in details.

3.3 Language Stack and Layers of Abstraction

DSL stacking [7] is a form of layering where higher-level, more specific DSLs are implemented on lower-level, more general DSLs. The concept framework is designed to enable DSL stacking in Semantic Web applications. See Fig. 3 for the language stack of the business process reasoning application.

Allegro Common Lisp is the base language in which AllegroGraph and AllegroProlog are implemented. The concept framework uses functionality of both libraries. Using the concept framework, a layered set of concepts is defined for the application domain, business process models: concrete modeling notations like UML activity diagrams and EPC diagrams on top of general graph based models on top of general Semantic Web concepts. Concrete reasoning applications can be implemented using those concepts.

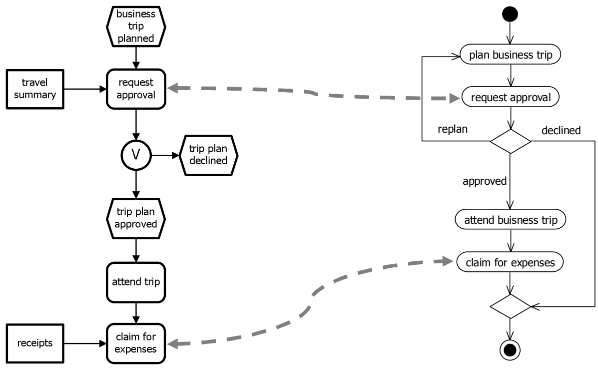


Fig. 2. Example business processes as EPC and UML activity diagram

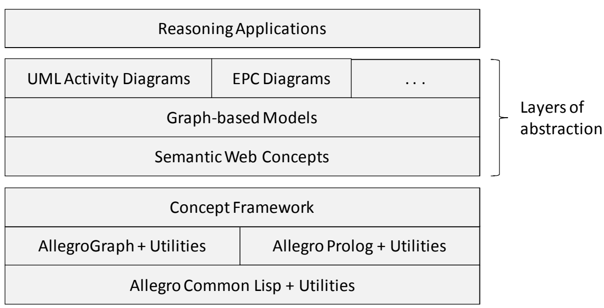


Fig. 3. Language stack

3.4 Concepts

Fig. 4 gives an overview of concepts being defined in the various layers of the business process reasoning application. One example concept, UML activity, is zoomed out.

UML activity is implemented using the general graph concept of a node. Node itself is implemented using the Semantic Web concept of an instance. Instance is implemented using triple from the concept framework.

The concept `activity` is defined as follows:

```
(define-concept activity(uri label diagram &optional
comment)
  (node uri !uml:activity diagram label comment))
```

With the concept `activity` defined on top of the concept `node`, the creation of a UML activity node can be expressed more concisely as in Section 2.4 as follows:

```
(activity !trv:req-app "request approval !trv:uml)
```

Using the concept `follows`, the edges of the UML activity diagrams can be asserted, e.g.,

```
(follows !trv:split !trv:req-app)
```

This code for asserting statements is typically generated, for example, from the XML output of a UML tool.

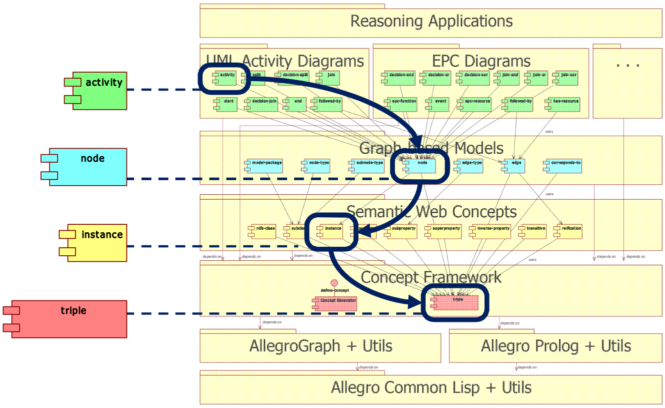


Fig. 4. Sample concepts

3.5 Querying and Reasoning

Asserted statements can conveniently be queried using the Prolog predicate generated from the concept specification – see the example in Section 2.4. In this section, we show how the concept framework supports the development of reasoning applications.

Determining the similarity between business process models is a complex task. We can rate similarity between two business process models concerning different aspects.

- *Diagram*: similarity of diagram titles and diagram types
- *Nodes*: similarity of node names and node types
- *Structure*: similarity of edge structures, e.g., similar nodes following other similar nodes

The following sample AllegroProlog rule detects a simple aspect of similarity between nodes, namely identical labels.

```
(<- (label-equivalence ?label ?n1 ?n2 ?d1 ?d2)
  (node ?n1 nil ?d1 ?label)
  (node ?n2 nil ?d2 ?label))
```

The rule consists of the *head term* `(label-equivalence …)` and the *goal terms* `(node …)`. If all goal terms can be proven then the head term is proven. The example reads like this: if two nodes `?n1` and `?n2` can be found in two diagrams `?d1` and `?d2` and their labels are identical (`?label`) then `label-equivalence` is true. For this predicate, the predicate `node` is used. It is satisfied no matter whether the nodes are UML activities, EPC functions or of any other node type. This is expressed by using `nil` as `node-type` parameter. The identity of the labels is assured via unification of the variable `?label`.

The following query selects all labels that occur as identical node names in any two diagrams.

```
(select ?label (label-equivalence ?label ?n1 ?n2 ?d1 ?d2))
```

Assuming that all nodes of the EPC diagram and the UML activity diagram in Fig. 2 have been asserted, the result is

```
("request approval" "claim for expenses")
```

When providing different parameters to the select statement, the following queries may be formulated using the predicate `label-equivalence`, all consisting of a single line of code in the query body:

1. Find equivalent labels in two specified diagrams
2. Instead of the labels, select the URIs of the nodes with identical labels in two diagrams
3. Select all diagrams in which there are identical node labels as in a given diagram
4. Select all pairs of diagrams with equivalent labels
5. Select all duplicates within one diagram
6. Select all duplicates within all diagrams

Many more types of queries may be expressed with this simple Prolog predicate.

We now present a more complex sample predicate to reason over structural similarity between diagrams.

```
(<- (succ-equivalence ?src1 ?src2 ?dest1 ?dest2 ?d1 ?d2)
  (label-equivalence ?lsrc ?src1 ?src2 ?d1 ?d2)
  (label-equivalence ?ldest ?dest1 ?dest2 ?d1 ?d2)
  (follows-transitively ?dest1 ?src1)
  (follows-transitively ?dest2 ?src2))
```

The predicate checks for successor equivalence, i.e., whether two pair wise equivalent nodes in two models follow each other – either directly or indirectly (`follows-transitively`). The possibilities of different meaningful queries using predicate `succ-equivalence` are even greater than the ones for `label-equivalence`. There is an enormous amount of different meaningful queries, each with a single line of code in the query body using the predicate `succ-equivalence`.

4 Evaluation

4.1 Comparison with Semantic Web Technologies

We now compare implementing ontologies and Semantic Web applications with and without using the concept framework.

For asserting statements consider, again, the example UML activity (Section 4.4). In N3, the equivalent assertion would be as follows:

```
trv:req-app rdf:type uml:activity;
            rdfs:label "request approval".
trv:uml modl:contains trv:req-app.
```

The assertion using the concept `activity` is more concise: 1 loc (line of code) compared to 3 loc in N3 notation, respectively 9 loc in N-triples notation, 15 loc in RDF/XML, and about 15-20 loc in the Semantic Web framework Sesame.

The concept `activity` hides implementation details: the use of class `uml:activity` and the use of properties `rdf:type`, `rdfs:label`, and `modl:contains`.

For comparing queries, consider the simple example of querying for label equivalence in Section 3.5. In SPARQL, the equivalent query would be as follows:

```
SELECT ?label WHERE {
  trv:uml modl:contains ?node1.
  trv:epc modl:contains ?node2.
  ?node-type1 rdfs:subClassOf modl:node.
  ?node1 rdf:type ?node-type1;
         rdfs:label ?label.
  ?node-type2 rdfs:subClassOf modl:node.
  ?node2 rdf:type ?node-type2;
         rdfs:label ?label.}
```

The WHERE part of the query comprises 8 lines of SPARQL code. In comparison, the WHERE part of the Prolog query using `label-equivalence` comprises 1 loc – with the definition of `label-equivalence` comprising 3 loc. Many other queries like, for example, selecting duplicates within diagrams (for more examples, see Section 3.5) may be formulated using `label-equivalence` – the WHERE part always being a one-liner. Using SPARQL, every single WHERE part has to be programmed individually and would comprise about 8 loc each time. Invoking the SPARQL query from a Semantic Web framework like Sesame adds about another 20 loc each.

Comparing more complex assertions and queries, the differences in code size even get larger. For example, one SPARQL query for the successor equivalence example from Section 3.5 comprises 16 loc compared to the one-liner in the Prolog query using `succ-equivalence` (definition: 5 loc). Varying the parameters passed, dozens of different Prolog queries may be formulated using `succ-equivalence` – with 1 loc each. Every equivalent SPARQL query would comprise about 16 loc (SPARQL) plus about 20 loc (Java) in the Semantic Web framework each.

4.2 Related Work

Many publications discuss the layering of Semantic Web languages (e.g., [5], [6]). Whereas those publications focus on the expressive power of the underlying languages and mechanisms, we take a Software Engineering view focusing on the ontologies and application code to be developed.

Humm and Engelschall introduce a method called DSL stacking [7]. According to the paradigm of Language-Oriented Programming, an application for a problem should be implemented in the most appropriate domain-specific language (DSL). DSL stacking is a method for implementing Language-Oriented Programming where DSLs are incrementally developed on top of each other thus providing layers of abstraction. The concept framework can be seen as a DSL stacking platform for the Semantic Web.

Knublauch introduces the SPARQL Inferencing Notation (SPIN, [8]). He also identifies the lack of abstraction mechanisms in Semantic Web technology. SPIN targets the same goals as the concept framework. It introduces so-called SPIN functions that are SPARQL functions which can be used in FILTER or LET statements. SPIN Templates are re-usable SPARQL queries that can be instantiated with parameters. SPIN templates can be used instead of typing in SPARQL queries by hand. Thus, SPIN functions and templates introduce named queries that can be invoked as subqueries which is not possible in the current SPARQL standard. The crucial difference to the concept framework is that asserting statements and querying are handled separately. In the concept framework, one concept definition may be used for asserting statements and reasoning / querying.

In [9], Eiter et al. present an approach for integrating rules and ontologies in the Semantic Web. The approach combines answer set programming with description logics. The rules being used are similar to Prolog rules with negation as failure. They may contain queries. Eiter et al. claim an encapsulation view that increases flexibility to be one advantage of their approach. Our approach of using Prolog rules in Semantic Web applications has a similar aim. It also allows for reasoning and querying. It fosters encapsulation and we also claim increased flexibility as a result. While Eiter et al. provide a sound conceptual basis for integrating rules and ontologies, however, they do not cover abstraction and encapsulation mechanisms for asserting statements.

In [10], Le-Pouc et al. identify limited software support and the lack of standard programming paradigms in Semantic Web standards. To alleviate those problems, they introduce Semantic Web Pipes to support fast implementation of semantic data mash-ups while preserving abstraction, encapsulation, component-orientation, code re-usability, and maintainability. They present piping operators including the CONSTRUCT and SELECT operators that allow using results of SPARQL queries to be used in further processing. We do agree with their analysis of the state-of-the-art in Semantic Web programming. Their solution has similarities with our solution in that higher-level, more specific named constructs (here: pipes) can use lower-level, more general constructs. Again, as with [8] and [9], the difference to our approach is the sole focus on querying, not on asserting – thus solely focusing on the Semantic Web application developer and not, additionally, on the ontology modeler.

5 Conclusions

We have presented a novel approach for realizing layers of abstraction in ontology modeling and Semantic Web applications. A concept framework allows specifying higher-level, more specific concepts on top of lower-level, more general concepts. From concept specifications, code for asserting statements (Lisp functions) as well as for reasoning and querying (Prolog predicates) is being generated. We have shown that using the concept framework considerably reduces code size of ontologies and Semantic Web applications. The amount of code savings can fairly be considered an order of magnitude. Using the concept framework furthermore enhances software quality regarding conciseness, understandability, and maintainability of the resulting code.

We have successfully used the concept framework in a complex application domain: reasoning over business process models. In this article, we have only shown a small and simple exemplary subset of the ontology and sample predicates for reasoning similarity. Additionally, we have implemented similarity metrics including linguistic analyses such as synonym resolution. In addition to similarity, we provide predicates for checking consistency of business process models and conformance to reference architectures. Our application not only covers the application scenario of mergers and acquisitions, but also the broader application scenario of e-business integration.

So far, our development is in the research and prototyping stage. Current and future work includes the following:

- Improving the stability of the concept framework
- Providing means for dedicated performance optimization of generated predicates where necessary
- Porting the concept framework to Semantic Web environments based on mainstream platforms like Java

We feel that our approach could, eventually, considerably improve the way ontologies and Semantic Web applications are developed in the future.

References

[1] Encyclopedia Britannica Ultimate Reference Suite (2010)

[2] Havey, M.: Essential Business Process Modeling. O'Reilly Media, Sebastopol (2005)

[3] van Deursen, A., Klint, P., Visser, J.: Domain-specific languages: an annotated bibliography. ACM SIGPLAN Notices 35, 26–36 (2000)

[4] Backus, J.W.: The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference. In: Proceedings of the International Conference on Information Processing, pp. 125–132. UNESCO (1959)

[5] Horrocks, I., Parsia, B., Patel-Schneider, P., Hendler, J.: Semantic Web Architecture: Stack or Two Towers?, pp. 37–41. Springer, Heidelberg (2005)

[6] Kifer, M., de Bruijn, J., Boley, H., Fensel, D.: A Realistic Architecture for the Semantic Web, pp. 17–29. Springer, Heidelberg (2005)

[7] Humm, B., Engelschall, R.: Language-Oriented Programming via DSL Stacking. In: Proceedings of the 5th International Conference on Software and Data Technologies (ICSOFT 2010), Athens, Greece, pp. 279–287 (2010)

[8] Knublauch, H.: The Object-Oriented Semantic Web with SPIN (2009), `http://composing-the-semantic-web.blogspot.com/2009/01/object-oriented-semantic-web-with-spin.html`

[9] Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining Answer Set Programming with Description Logics for the Semantic Web. In: Proc. KR 2004. AAAI Press (2004)

[10] Phouc, D.L., Polleres, A., Morbidoni, C., Hauswirth, M., Tummarello, G.: Rapid Prototyping of Semantic Mash-Ups through Semantic Web Pipes. In: Proceeedings of the 18th International World Wide Web Conference (WWW 2009). ACM, Madrid (2009)

Onto-Ann: An Automatic and Semantically Rich Annotation Component for Do-It-Yourself Assemblage

Yan Tang

VUB STARLab
10G731, Vrije Universiteit Brussel
Pleinlaan 2, 1050 Elesene
Brussels, Belgium
yan.tang@vub.ac.be

Abstract. In a Do-It-Yourself software assemblage environment, it is important for the amateurs and technicians to find the right building blocks before assembling their own solutions. We have designed an ontology-based do-it-yourself architecture, which assists users to find suitable components and guide them to the assemblage. In particular, a tool called DIY-CDR (Do-It-Yourself Component Discover and Recommender) has been designed and implemented. The matching engine in DIY-CDR uses domain ontologies and annotation sets of the components and compares users' requirements to the annotation sets. Since the components contain little metadata information and their descriptions are often free texts, how to automatically annotate these components becomes a problem. In this paper, we propose a solution called Onto-Ann, which is an automatic and semantically rich annotation tool. It uses combined technologies from natural language processing (NLP), social study and ontology.

Keywords: Ontology, Natural Language Processing (NLP), Internet-of-Things (IoT).

1 Introduction

Internet-of-Things (IoT) enables uniquely identified tangible objects to be represented in a web structure and further communicate with each other using modern ontology engineering technologies. In the EU ITEA-2 Project "Do-It-Yourself Smart Experiences"[1] [13], a challenge is to provide possibilities to user-generated IoT applications. It aims at applying the freedom of creativity of Web 2.0 or Web 3.0 to the IoT. Essentially, non-expert users should be able to search for public devices (or software modules) or share their own, privately bought or built devices. As a result, they can personalize their own smart environment.

A do-it-yourself (DIY) component discoverer and recommender (DIY-CDR, [17]) has been designed to assist users to find proper components before he starts his DIY

[1] http://dyse.org:8080 (last retrieved on July 10th, 2011).

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 424–433, 2011.
© Springer-Verlag Berlin Heidelberg 2011

processes. DIY-CDR uses ontology-based matching strategies as its kernel component for the discovery tasks. It takes a context ontology, domain ontologies, component annotations and a user query as the input, and generates a list of components as the output.

We are currently facing a problem while using DIY-CDY. That is, there are too many components (including hardware and software components). To manually annotate them with the context ontology and domain ontologies becomes almost impossible. Therefore, we need to design an automatic annotation method.

How to automatically annotate hardware components (e.g. sensors and actuators) is easy to tackle as long as these components are connected to the same cloud, follow same inference/description formats (e.g., WSDL[2] and JSON[3]) or use same web services (e.g. SOAP[4] and REST[5]).

This paper focuses on how to automatically annotate software components when their meta-information is limited but their descriptions are in rich, human readable and free texts. Our approach is called Onto-Ann. Although Onto-Ann is designed, implemented and tested for do-it-yourself assemblage, we observe that it is not limited to this particular domain/application. It can be used as an ontology-based annotation tool in general.

The paper is organized as follows: Sec. 2 is the paper background and related work. We describe Onto-Ann in Sec. 3. In the meanwhile, we present its implementation. We illustrate some discussions, our future work and conclude the paper in Sec. 4.

2 Background and Related Work

The definitions of annotation can be found in several literatures. An annotation (more specifically, *semantic annotation*) is additional information that identifies or defines a concept in a semantic model in order to describe part of that document [4].

The authors in [8] define semantic annotation as a way of assigning to the entities in the text that links to their *semantic descriptions*. It is considered as a kind of metadata, which provides both *class* and *instance* information about the entities.

The authors in [12] have a quite simple view on semantic annotation. They treat semantic annotation as a task to *tag* ontology class instance data and map it into ontology classes.

In ONTOTEXT[6], annotation is also considered as *tagging* (yet not exactly tagging), which is about attaching names, attributes, comments, descriptions, etc. to a document or to a selected part in a text. Compared to tagging, which speeds up

[2] `http://www.w3.org/TR/wsdl` (last retrieved on July 10th, 2011)
[3] `http://www.json.org/` (last retrieved on July 10th, 2011)
[4] `http://www.w3.org/TR/soap/` (last retrieved on July 10th, 2011)
[5] `http://www.w3.org/Submission/SA-REST/` (last retrieved on July 10th, 2011)
[6] `http://www.ontotext.com/kim/semanticannotation.html` (last retrieved on July 10th, 2011)

searching and helps end users to find relevant and precise information, semantic annotation goes one level deeper: 1) it enriches the *unstructured* or *semi-structured* data with a *context* that is further linked to the structured knowledge of a domain. 2) It allows results that are not *explicitly* related to the original search. So, if tagging is about promptly finding the most relevant result, semantic annotation adds diversity and richness to the process.

The semantic annotation technology enables many new applications, such as highlighting, indexing and retrieval, categorization, structuring free texts and metadata generation. It is applicable for any kinds of texts – web pages, office documents, descriptions of a data table field, meta-information of a structured resource, additional comments for a document and so forth.

There exist several approaches to semantic annotations. SAWSDL [4] provides semantic annotations, which are XML attributes added to a WSDL or associated XML Schema document, at the XML element they describe. It contains two kinds of semantic annotations: 1) explicit identifiers of concepts, and 2) identifiers of mappings from WSDL to concepts or vice versa.

KIM [8] is a system allowing annotation, indexing, and retrieval of documents from the web with respect to real-world entities. It is based on a simple upper-level ontology and domain specifications that are built upon this ontology (called "KIMO"). It allows dynamic and user-specific semantic annotations.

GoNTogle [1] is a framework for document annotation and retrieval, built on top of Semantic Web and information retrieval (IR) technologies. Automatic annotation is based on a learning method that exploits user annotation history and textual information to automatically suggest annotations for new documents. A flexible combination of keyword-based and semantic-based search over documents is proposed in conjunction with advanced ontology-based search operations.

The authors in [6] present an approach to automating semantic annotation within service-oriented architectures. They consider the annotation process as the task of matching an arbitrary word or term with the most appropriate concept in the domain ontology. Their approach to semantic annotation uses text mining technologies and Google Distance. The main contribution is to propose a term matching algorithm based on text mining and IR.

Reeve and Han [12] examine current Semantic Web annotation platforms that provide annotation and related services, and review their architecture, approaches and performances. Including KIM [8], platforms like AeroDAML [9], Armadillo [3], Ont-O-Mat [7], SemTag [2], MnM [18] and MUSE [11] are discussed and compared.

Compared to the related work, our main contribution is to propose a method that combines natural language processing and social study. We will discuss how experts from different communities can contribute to configure the semantic annotation algorithm, which we call Onto-Ann, and provide knowledge as its input.

3 Onto-Ann

Onto-Ann is designed as illustrated in Fig. 1.

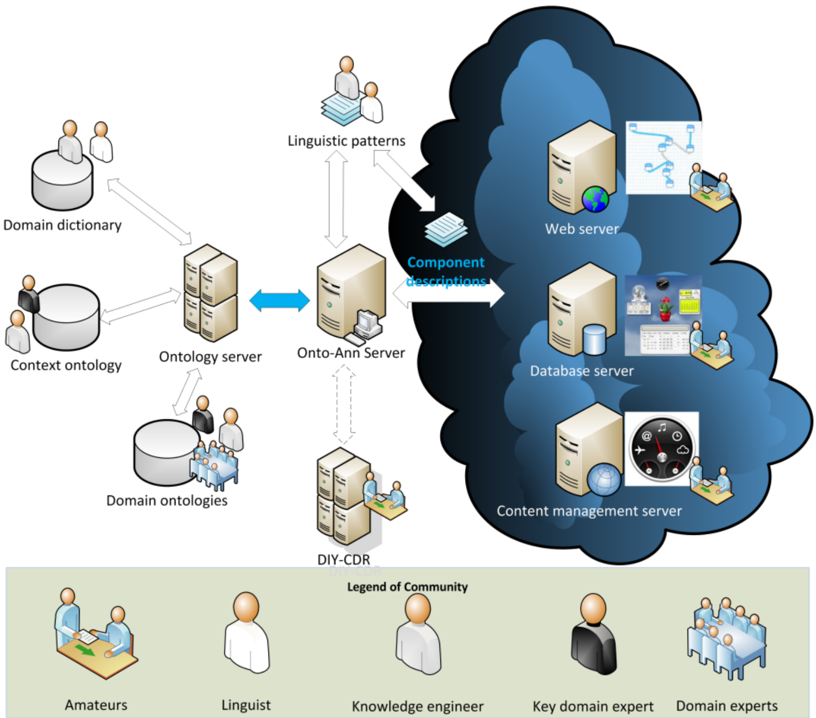


Fig. 1. Design of Onto-Ann

Onto-Ann involves people playing five roles:

- **Amateurs** (semi-professionals), who are the users of DIY-CDR. They have the needs of discovering existing software components. After they have created their own solutions, they want to share them with others using community portals. These community portals can be web sites of interests, gadgets databases or linked data.
- **Domain experts**. They are responsible for providing knowledge concerning domain ontologies to the key domain expert. They are involved when the domain ontologies need to evolve over time.
- **Key domain expert**. This person decides which part of domain knowledge is essential to build the domain ontologies and which part is explanatory text. He has an excellent overview of the whole domain and can grasp fast the focus of the ontologies of the current version. As not all the knowledge provided by the domain experts is equally important, he is responsible for filtering out less important knowledge and passing the most essential parts to the knowledge engineer.

- **Knowledge engineer**, who is an expert in formalizing knowledge into ontologies, needs to help the key domain expert for finalizing the ontologies. He knows very well how to model ontologies, store and publish them.
- **Linguist**, who analyzes the textual descriptions of the software components, is responsible to create, check and update the linguistic patterns that are used for the annotation tasks. Note that these linguistic patterns must comply with ontologies. Therefore, the knowledge engineer should assist the linguist during this process. They are also responsible for managing the terminology in the domain dictionary.

As shown in Fig. 1, the Onto-Ann structure contains the following six servers.

- **Web server**, which is a web crawler or online document scratcher, finds software components and collects their textual descriptions. These software components can be, for example, Yahoo Pipes[7] and Google Gadgets[8].
- **Database server**, which is a domestic server, stores the textual information, structural model, metadata and schemata of the software that is used within an enterprise or a user community, e.g. Pure Data Patches[9].
- **Content management server**, which is a personal server, stores the information or meta-information of personal software components. These personal software components can be, for instance, personal data stored in the Android[10] applications in a personal smart phone.
- **Ontology server**, which stores, queries and updates ontologies, communicates with the databases that store domain ontologies, a context ontology and a domain dictionary. As an example, our domain ontology is the Yahoo Pipe ontology, which will be illustrated. In our experimental settings, the domain dictionary contains WordNet [5] and a fact type map, which will be illustrated later in this section.
- **DIY-CDR**, which uses the annotations generated by Onto-Ann, discovers suitable software components for the end users. We refer to [17] for the details.
- **Onto-Ann Server** is a server that takes as inputs the ontologies, domain dictionary, linguistic patterns, and textual descriptions of the components. It generates a list of annotations in the format of binary fact types.

We use the Developing Ontology-Grounded Methods and Applications framework (DOGMA, [14]) for modeling our ontologies. An ontology modeled in DOGMA has two layers: a lexon layer and a commitment layer.

A lexon is a simple binary fact type, which contains a context identifier, two terms and two roles. For instance, a lexon $\langle \gamma, teacher, teaches, is\ taught\ by, student \rangle$ presents a fact that "a teacher teaches a student, and a student is taught by a teacher", where "teacher" and "students" are two terms, "teaches" and "is taught by" are two

---

7 `http://pipes.yahoo.com/pipes/` (last retrieved on July 10th, 2011).
8 `http://www.google.com/ig/directory?synd=open` (last retrieved on July 10th, 2011).
9 `http://puredata.info/community/patches` (last retrieved on July 10th, 2011).
10 `http://www.android.com/` (last retrieved on July 10th, 2011).

roles. The context identifier γ points to a resource where “teacher” and “student” are originally defined and disambiguated.

A commitment (also called “ontological commitment”) is an agreement made by a community (also called “group of interests”). This community in this paper is the group that contains domain experts and a key domain expert (see Fig. 1). A commitment is a rule in a given syntax. For instance, we can use the decision commitment language (DECOL, [16]) to write a commitment shown as follows.

```
P = [Student, takes, is taken by, Exam]: MAND (P).
```

The above commitment contains a mandatory constraint, which means that each student takes at least one exam.

As discussed, we have two kinds of ontologies. One is a context ontology [17], which in some cases called “upper ontology” or “upper level ontology”. The other is domain ontologies, e.g. Yahoo Pipe ontology [17]. The models in domain ontologies should follow the patterns that are derived from this context ontology. The pattern-driven ontology creation methodology (PAD-ON, [15]) can be used to derive domain ontologies from the context ontology.

We use binary fact types (lexons) to model the elements in an ontology. A *fact type map* contains triples $\langle t, r_a, t' \rangle$ where t is a textual object in a sentence (e.g. a noun), t' is a lexon term in the ontology and r_a ($r_a \in R_a$) is an annotation relation between t and t'. A fact type map also contains triples $\langle r, r_b, r' \rangle$ where r is a verb in a sentence that can play as a role, r' is a lexon (co-)role defined in the ontology and r_b ($r_b = synonym$) is an annotation relation between r and r'.

$$R_a = \{synonym, has\ label, has\ tag, is\ an\ instance\ of, has\ type, has /is\ of, equivalent\ to\}$$

The lexon (co-)roles are not freely defined in our ontology. We only allow domain canonical relations and annotation relations as lexon (co-)roles. A domain canonical relation is a (co-)role in the context ontology or in domain ontologies. In our example, the domain canonical relation set contains the relations of “is member”/“has member”, “is measured with”/“measure”, “execute”/“is executed by”, “is a”/“supertype of”, “use”/“is used by”, “is”, “execute task during”/ “task is executed by during”, “use object at”/ “object is used by at”, and “has”/“is of”.

Table 1. Lexon role pairs and their source

Relation (lexon role pair)	Source
Synonym/synonym	WordNet Synset
Has label/is label of	Meta-information of a webpage
Has tag/is tag of	Meta-information of a webpage
Is an instance of/has instance	Local database and linked data
Has type/is type of	Local database and linked data
Has/is of	Domain ontologies, WordNet, domain dictionary
Equivalent to/equivalent to	Domain ontologies and domain dictionary
Domain canonical relations	Domain ontologies, linguistic patterns and POS

When we get the fact type map, we need to transform the triples into lexons. We add a co-role and a context identifier to each generated annotation triple. The role and co-role pair is illustrated in the column "Relation (lexon role pair)" in Table 1. The context identifier points to its source, which shows how this relation is derived.

Note that the domain canonical relations (Table 1.) are derived from domain ontologies. Linguistic patterns and Part-Of-Speech (POS) taggers help us to filter unnecessary textual information and building lexons.

Table 2. Title "Add Feed Label to Each Item Title"
URL: http://pipes.yahoo.com/pipes/pipe.info?_id=xjMFQsOG3BGSovq98SvLAg

…I like to add a tag or description of the blog…~~Note: There's a problem with my regex and some non-standard characters tend to muck up the title.~~ ~~If anyone has any comments on how to fix it, let me know.~~

For instance, we have a textual description of a Yahoo Pipe as shown in Table 2. We use the following regular expression to filter unnecessary information (see the texts that are stroked though in Table 2).

```
Note: [A-Za-z' -.]+
```

Table 3. A sentence from Table 2 tagged using Penn Tree Bank POS tagger [10]

I/PRP like/VBP to/TO add/VB a/DT tag/NN or/CC description/NN of/IN the/DT blog/NN

We tag the selected textual information using Penn Tree Bank POS tagger [10]. The meanings of the taggers shown in Table 3 are illustrated as follows.

- PRP – pronoun, personal
- VBP – verb with present tense
- TO – "to"
- VB – verb in a base form
- DT – determiner
- NN – noun
- CC – coordinating conjunction
- IN – preposition or conjunction, subordinating

For the tags that are not illustrated above, we refer to [10] and the online resources[11] for its details. Based on the tags, we can get a set of triples containing "subject-verb-object". We transform the example shown in Table 3 in the method that is illustrated in Fig. 2.

If we can find "subject-verb-object" as a lexon $\langle \gamma, subject, verb, co-role, object\rangle$ or $\langle \gamma, object, role, verb, subject\rangle$ in the ontologies, then we can use this lexon as its annotation. If not, then we need to have at least three lexons. The first one is $\langle \gamma, Person, execute, is\ executed\ by, verb\rangle$. The second one is $\langle \gamma, Person, use, is\ used\ by, Object\rangle$. The third one is a lexon that contains at least one lexon role pair in Table 1 and one concept (lexon term) in the ontologies. Note that in this

[11] `http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html` and `http://www.computing.dcu.ie/~acahill/tagset.html` (last retrieved on July 12, 2011).

example, the pronoun "I" is translated into "Person". Besides, we also consider other personal pronouns such as "you", "he" and "they" as "Person". Note also that if a sentence starts directly with a verb, then we also assume that the subject is "Person".

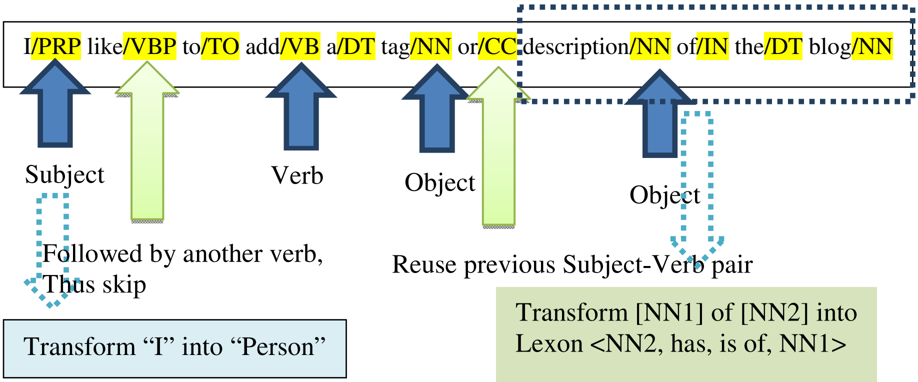


Fig. 2. Transform tagged texts into annotated lexons

When we see a linguistic pattern like "$[NN_1]\ of\ [NN_2]$", then this phrase need to be translated into a lexon $\langle \gamma, NN_2, has, is\ of, NN_1 \rangle$.

Using the method illustrated in Fig. 2, we can derive the following lexons.

$$\langle \gamma, Person, add, is\ added\ by, Tag \rangle$$
$$\langle \gamma, Person, use, is\ used\ by\ by, Description \rangle$$
$$\langle \gamma, Person, execute, is\ executed\ by, Add \rangle$$
$$\langle \gamma, Blog, has, is\ of, Description \rangle$$

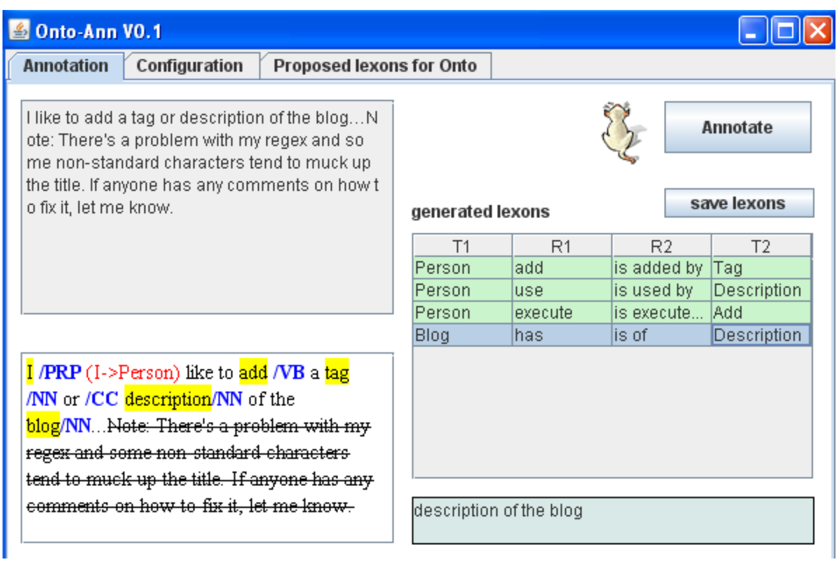


Fig. 3. Onto-Ann screenshot

We use the Stanford Part-of-Speech (POS) Tagger, which is a Java implementation of a maximum-entropy (CMM) part-of-speech (POS) tagger[12], to discover verbs and nouns in a sentence.

We use MySQL Server 5.0 as the local database server and DBpedia[13] as the linked data server to query data from Wikipedia. The first prototype has been implemented as a standalone Java Swing application. Its screenshot is shown in Fig. 3.

4 Discussion, Conclusion and Future Work

In this paper, we have discussed an automatic ontology annotation method, which we call Onto-Ann. Compared to the related work, our main contribution is to propose a method that combines natural language processing, ontology engineering and social study. Different kinds of experts from different communities can contribute to configure Onto-Ann and provide knowledge as its input.

We have implemented the initial design of Onto-Ann. It supports annotation using existing concepts in the domain ontologies. With regard to its performance, the average cost in time for invoking Stanford POS tagger and process the example shown in this paper is 3422 milliseconds. If a large text needs to get parsed, then it gets naturally worse. Just like many NLP programs, the performance is indeed a problem.

We observe that it is possible to add "new" concepts that are discovered during such annotation processes. How to propose these new concepts to existing ontologies, and help domain experts in ontology versioning could be an interesting challenge and future work. We need to keep in mind that, when ontologies evolve, the annotation should evolve as well.

We are currently working on a method to judge the quality of the resultant annotations of Onto-Ann.

Acknowledgments. The work has been supported by the EU ITEA-2 Project 2008005 "Do-it-Yourself Smart Experiences", founded by IWT 459.

References

[1] Bikakis, N., Giannopoulos, G., Dalamagas, T., Sellis, T.: Integrating Keywords and Semantics on Document Annotation and Search. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6427, pp. 921–938. Springer, Heidelberg (2010), doi:10.1007/978-3-642-16949-6_19

[2] Dill, S., Gibson, N., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In: Twelfth International World Wide Web Conference, Budapest, Hungary, pp. 178–186 (2003)

[12] `http://nlp.stanford.edu/software/tagger.shtml` (last retrieved on July 10th, 2011)

[13] `http://dbpedia.org` (last retrieved on July 12th, 2011)

[3] Dingli, A., Ciravegna, F., Wilks, Y.: Automatic Semantic Annotation using Unsupervised Information Extraction and Integration. In: K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation (2003)

[4] Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema, W3C Recommendation, August 28 (2007), `http://www.w3.org/TR/sawsdl/`

[5] Fellbaum, C.: WordNet: an electronic lexical database. Language, Speech, and Communication. MIT Press, Cambridge (1998) ISBN-10:026206197X, ISBN-13: 978-0262061971

[6] Grčar, M., Klien, E., Novak, B.: Using Term-Matching Algorithms for the Annotation of Geo-services. In: Berendt, B., Mladenič, D., de Gemmis, M., Semeraro, G., Spiliopoulou, M., Stumme, G., Svátek, V., Železný, F. (eds.) Knowledge Discovery Enhanced with Semantic and Social Information. SCI, vol. 220, pp. 127–143. Springer, Heidelberg (2009)

[7] Handschuh, S., Staab, S., Ciravogna, F.: S-CREAM - Semi-automatic CREAtion of Metadata. In: SAAKM 2002 -Semantic Authoring, Annotation & Knowledge Markup - Preliminary Workshop Programme (2002)

[8] Kiryakov, A., et al.: Semantic annotation, indexing, and retrieval. Journal of Web Semantics 2, 49–79 (2004)

[9] Kogut, P., Holmes, W.: AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In: First International Conference on Knowledge Capture (2001)

[10] Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the penn treebank. In: Computational Linguistics - Special Issue on Using Large Corpora: II, vol. 19(2). MIT Press, Cambridge (1993)

[11] Maynard, D.: Multi-Source and Multilingual Information Extraction. In: Proc. of BCS-SIGAI Workshop, UK, September 12 (2003)

[12] Reeve, L., Han, H.: Survey of semantic annotation platforms. In: Proc. of the 2005 ACM Symposium on Applied Computing, pp. 1634–1638. ACM, New York (2005) ISBN:1-58113-964-0

[13] Roelands, M., et al.: The DiY Smart Experiences Project. In: Uckelmann, D., et al. (eds.) Architecting the Internet of Things, ch. 13, Springer, Heidelberg (2011) ISBN 978-3-642-19156-5

[14] Spyns, P., Tang, Y., Meersman, R.: An Ontology Engineering Methodology for DOGMA. Journal of Applied Ontology, Special Issue on Ontological Foundations for Conceptual Modeling 3(1-2), 13–39 (2008); Guizzardi, G., Halpin, T. (eds.)

[15] Tang, Y., De Baer, P., Zhao, G., Meersman, R.: On constructing, grouping and using topical ontology for semantic matching. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 816–825. Springer, Heidelberg (2009); The 5th International IFIP Workshop on Semantic Web and Web Semantics (SWWS 2009) ISBN: 978-3-642-05289-7

[16] Tang, Y.: Semantic Decision Tables - A New, Promising and Practical Way of Organizing Your Business Semantics with Existing Decision Making Tools. LAP LAMBERT Academic Publishing AG & Co. Saarbrucken, Germany (2010) ISBN 978-3- 8383-3791-3

[17] Tang, Y., Meersman, R.: DIY-CDR: An Ontology-based, Do-it-Yourself Components Discoverer and Recommender. Journal of Personal and Ubiquitous Computing (to appear, 2011); Yu, Z., Cheng, D., Khalil, I., Kay, J., Heckmann, D. (eds.), doi:10.1007/s00779-011-0416-y;

[18] Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., Ciravegna, F.: MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 379–391. Springer, Heidelberg (2002)

Ontology to Represent Similarity Relations between Public Web Services

Maricela Bravo

Systems Department, Metropolitan Autonomous University (UAM)
Av. San Pablo No. 180, Col. Reynosa Tamaulipas, Delegación Azcapotzalco
CP. 02200, Distrito Federal, México
mcbc@correo.azc.uam.mx

Abstract. Currently Internet is largely populated with Web services offered by different providers and published in various Web repositories. However, public available Web services still suffer from problems that have been widely discussed, such as the lack of functional semantics. This lack of semantics makes very difficult the automatic discovery and invocation of public Web services, even when the system integrator can obtain a copy of the WSDL file. This paper describes an ontological approach for discovering similarity relations between public Web services. The objective of this work is to extract relevant data that is coded into service descriptions, calculate similarity measures between them, represent discovered similarities in an ontological form, and execute inference. Experimental results show that the overall process towards the automation of public Web services discovery based on ontology population and structural similarity measures is feasible.

Keywords: Web Services, Structural Similarity Measures, Similarity Relations Discovery, Automated Ontology Population, Inference.

1 Introduction

Currently Internet is largely populated with Web services offered by different providers and published in various Web repositories. Interface descriptions of these services are easily available mostly in WSDL files, allowing system developers and integrators to semi-automatically construct proxy clients that invoke service operations. However, public available Web services still suffer from problems that have been largely discussed. In 2004, Dong et.al [1] stated that texts fragments included in Web service descriptions were insufficient for keyword search and the underlying structure and semantics of Web services were not exploited. Regarding Web service composition Jianchun and Subbarao [2] found very few ways of composing public available Web services, mainly because of the lack of services and correlations among them. More recently, Rodríguez et.al [3] identified various common mistakes in WSDL documents: inappropriate or lacking comments, use of ambiguous names for the main elements, redundant port-types, low cohesive operations in the same port-type, enclosed data models, redundant data models, among others.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 434–443, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Current solution approaches rely on semantic annotations (SAWSDL[1]) and ontological models (OWL-S[2] and WSMO[3]). However, these solutions have added requirements for developers and integrators. For instance, the provider of Web services has to understand and select the ontology-based language together with the set of tools in order to build an ontological model to describe and offer his services. On the other hand, the client has to recognize the service description language in use (WSDL, OWL-S, SAWSDL or WSML), and use the particular search mechanisms and programming libraries to find providers into public repositories.

The main objective of this work is to build a new ontology to represent services, enable the representation of similarity relations, define inference rules to derive new relationships, and facilitate representation of other service description languages.

It is clear that the use of ontology languages and the use of reasoning tools are strongly marking the continuance of these technologies in the near future. Moreover, it is important to note that even though Web services are described using different languages, there are common elements that all Web services description languages share: a *communication interface* that the client uses to invoke the service remotely. This communication interface must describe information about the *functions* that the service offers (*operation* in WSDL, *profile* in OWL-S or *capability* in WSMO) as well as the correct description of *input* and *output* parameters.

The rest of the paper is organized as follows: in Section 2, related work is presented; in Section 3 the design of the ontology is described; in Section 4, similarity measures for relationships discovery are presented; in Section 5 the experimental architecture and results are presented; and conclusions in Section 6.

2 Related Work

At the core of many complex Web service tasks there are similarity measures supporting them. For instance, to support discovery and matchmaking, there is the need to compare a given service specification against the set of published services; similarly occurs when searching and selecting services for composition. In this section, a classification of similarity measures is described.

a) Syntactic and Structural

Syntactic similarity measures are those based on lexical comparisons, such as a string to string comparison. Whereas structural similarity measures are those that exploit signature or interface information of the service, such as: input parameters, output parameters, function names and descriptions, and service name and descriptions. It is a common task that a combination of a syntactic and structural approach is applied to support many service related tasks, like: classification, clustering, or discovery among others.

Some well known distance measures used in information theory to compare strings are the Jaccard coefficient [4], the Hamming distance [5]; and the Levenshtein

[1] `http://www.w3.org/2002/ws/sawsdl/`
[2] `http://www.w3.org/Submission/OWL-S/`
[3] `http://www.w3.org/Submission/WSMO/`

distance [6] which sets the smallest number of substitutions, insertions, and deletions of symbols that can be used to transform one string into another. In 2004, Dong et.al [1] described a Web service search engine, called Woogle[4]. They used a set of 431 public available services, applied similarity measures between operations, input and output parameters. A similar approach was implemented by Jianchun and Subbarao [2] who presented a study of a set of 2432 retrieved public Web services, applied a Hierarchical Agglomerative Clustering (HAC) algorithm and the Jaccard similarity as the distance measure between service descriptions. They found that most public Web services were simple data sources for data sensing and conversion. Regarding Web service composition they found very few ways of composing public available Web services, mainly because of the lack of services and correlations among them.

The UDDI Registry By Example (URBE) for Web service retrieval uses the evaluation of similarity between Web service interfaces [7]. The algorithm combines the analysis of the interfaces structures and the analysis of the terms used inside them. The higher the similarity, the less are the differences among their interfaces defined with Web Service Description Language (WSDL). URBE is useful to find a Web service suitable to replace an existing one. This is relevant for autonomic systems in order to ensure auto-management and auto-configuration. Semantic Annotation for WSDL (SAWSDL) is adopted as a language to annotate a WSDL description. The URBE prototype extends the Universal Description, Discovery and Integration (UDDI) compliant Web service registry.

Elgazzar et.al [8] use the Quality Threshold (QT) clustering algorithm to cluster WSDL documents using five key features: WSDL content, which is extracted and processed to build a vector of meaningful content words for each Web service; WSDL Types, extracted to determine the number of type matches between Web services; WSDL Messages, which encompass one or more logical parameters; WSDL Ports, which defines the combination and sequence of messages for an operation; and Web service Name.

b) Semantic

Semantic similarity measures use dictionaries or semantic references such as ontologies to construct representations of concepts and meanings. Bruno et.al [9] describe a semantic-based approach for automated classification and annotation of Web service description files (WSDL). Their approach relies on Support Vector Machines (SVM) and Information Retrieval Vector Spaces for service classification. Stroulia and Wang [10] developed and evaluated three methods to assess the similarity between two WSDL specifications. The first method is based on the Vector-space model information retrieval and WordNet considering words at the lexical level only. The second method represents an extension of the signature-matching method for component retrieval; this method is a structural-based approach. And the third method combines the structure-based matching with a semantic approach, by using WordNet to calculate the semantic distances between each pair of compared elements in the WSDL specifications.

The semantic Web has influenced many works by providing logic-based mechanisms to describe, annotate and discover Web services. Within this context,

[4] `http://db.cs.washington.edu/woogle.html`

McIlarith, Cao Son & Zeng [11] proposed one of the first initiatives to markup Web services based on DAML (ontology language), which started the important research area of "Semantic Web Services". The term Semantic Web Services is related to the set of frameworks and technologies which incorporate design and implementation of ontologies as a mechanism to enhance or annotate semantically service descriptions, for instance OWL-S, WSMO, and SAWSDL. Bener [12] proposed a matchmaker architecture which performs semantic matching of Web services on the basis of input and output descriptions of semantic Web services as well as precondition and effect matching. Klush [13] presented WSMO-MX service matchmaker which applies different matching filters to retrieve semantic web services written in WSML.

c) Behavioral

These measures attend the need to search and discover a Web service based on observable behavior. A functional quality of service approach to discover and compose interoperable Web services is described in [14]. They consider as functional attributes the service category, the service name, the operation name, the input and output messages and the annotation of the service.

Grigori, Corrales & Bouzeghoub [15] address the problem of service behavioral matching by implementing a graph matching algorithm which is based on the edit distance similarity measure. Similarly, Dijkman, Dumas & García-Bañuelos, [16] compare four graph matching algorithms to discover business process similarity. Nejati et.al [17] describe Match and Merge operators for Statecharts, their Match operator is based on typographic and linguistic similarities between the vocabularies of the different models producing a corresponding relation between the states of compared models; their Merge operator produces a merge that: preserves the behavioral properties of the models, respects the hierarchical structures, and distinguishes between shared and non-shared behaviors of compared models.

Measures based on syntax and structure provide information related to the communication interface of the service, but say nothing about the functional and behavioral information. On the other hand, semantic approaches strongly depend on human intervention, because there is always the requirement to manually annotate or enhance semantically Web services, providing their input, output, preconditions, and effects (IOPE) annotations.

3 Ontology to Represent Similarity Relations

In 1993 Gruber defined Ontology as "an explicit specification of a conceptualization" [18]. The Web service ontology described in this paper aims at providing fundamental inter-relations representation of Web service core concepts (*functions*, *input parameters* and *output parameters*). The main entities (classes) are *Service*, *Function* and *Parameter*. Figure 1 shows the diagram of this ontology and its interrelationships. The *Parameter* class is sub-classified into *InputParameter* and *OutputParameter*.

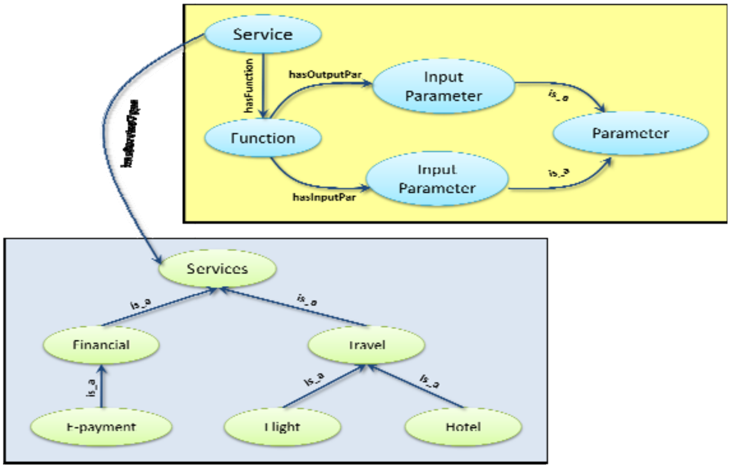


Fig. 1. Ontology diagrams for service representation and service application domain

Data type properties were defined as follows. For class *Service hasServiceName* and *hasURL* data type properties were defined to take only *xsd:string* data values. For class *Function*, the *hasFunctionName* data type property was established, allowing only *xsd:string* data values. For class *Parameters*, the *hasParameterDataType* and *hasParameterName* data type properties were defined to take *xsd:string* data values. Identification of semantic relationships between individuals of different classes in the ontology is implemented as object properties. *Service* class relates with *Function* class through *hasFunction* object property. Restrictions to this property are that a service has at least one function, and can have many functions. *Function* class is related with class *Parameter* through the *hasInputParameter* and *hasOutputParameter* object properties. Object property *hasInputParameter* is restricted to take values only from the class *InputParameters*; likewise, *hasOutputParameters* object property takes values only from the *OutputParameters* class.

4 Similarity Measures

Based on the work reported in [19], in this section various structural similarity measures are described.

Function Name Similarity. Let $Oname_1$ and $Oname_2$ be two compound function names from different Web services. $Oname_1$ consisting of a set of lexical tokens identified by $OnameTokens_1$. $Oname_2$ consisting of a set of lexical tokens identified by $OnameTokens_2$. The name lexical similarity between them is calculated with the Jaccard Similarity Coefficient:

$$FunctionNameSim(Oname_1\,,\,Oname_2) = \left|OnameTokens_1 \cap OnameTokens_2\right| \,/\, \left|OnameTokens_1 \cup OnameTokens_2\right| \quad (1)$$

The *FunctionNameSim* similarity measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity between both function names, and a returned value of 0 represents a total difference between names.

Input Parameter Similarity. Let $O_1 = (Oname_1, Ip_1)$, $O_2 = (Oname_2, Ip_2)$ be two functions from different Web services, with $Oname_i$ representing the function name and Ip_i the set of n input parameters described as follows:

$Ip_1 = \{ (nameP_1, typeP_1), (nameP_2, typeP_2), \ldots , (nameP_n, typeP_n)\}$,

$Ip_2 = \{ (nameP_1, typeP_1), (nameP_2, typeP_2), \ldots , (nameP_n, typeP_n)\}$.

Each parameter is defined by a pair of name and data type *(nameP, typeP)*. The input parameter similarity is calculated as follows:

$$InputParSim(O_1 , O_2) = |Ip_1 \cap Ip_2| \;/\; |Ip_1 \cup Ip_2| \tag{2}$$

The *InputParamSim* measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity, and a value of 0 represents a total difference.

Output Parameter Similarity. Similarly to Formula 1, a measure to evaluate the lexical similarity between output parameter names is defined. Let $OPname_1$, $OPname_2$, be two output parameter names from different Web service functions, each consisting of a set of lexical tokens identified by $OPnameTokens_1$ and $OPnameTokens_2$, respectively. The output parameter name lexical similarity is calculated by:

$$OPnameSim(OPname_1 , OPname_2) = |OPnameTokens_1 \cap OPnameTokens_2| \;/\; |OPnameTokens_1 \cup OPnameTokens_2| \tag{3}$$

The *OPnameSim* measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity, and a value of 0 represents a total difference.

Let $OPtype_1$, $OPtype_2$, be two output parameter data types from different Web service functions. The output parameter data type similarity between them is calculated as follows:

$$OPtypeSim(OPtype_1 , OPtype_2) = \begin{cases} 1, & \text{if } OPtype_1 = OPtype_2 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The *OPtypeSim* measure will return a value of 1 if both types are equal, and a value of 0 if they are different.

Average Output Similarity. Let $O_1 = (Oname_1, Op_1)$, $O_2, = (Oname_2, Op_2)$, be two functions from different Web services, with name $Oname_i$ and the output parameter object Op_i of function i. Each output parameter object Op_i consists of a pair of name and data type, $Op_1 = (OPname_1, OPtype_1)$, and $Op_2 = (OPname_2, OPtype_2)$. Particularly, O_1 and O_2 are output equivalent if $((OPname_1 = OPname_2)$ **and** $(OPtype_1 = OPtype_2))$.

The output parameter similarity is calculated as the average of output parameter name similarity and output parameter data type similarity.

$$OutputParSim(O_1 , O_2) = [\, OPnameSim(OPname_1 , OPname_2) + OPtypeSim(OPtype_1 , OPtype_2)\,] \,/\, 2 \tag{5}$$

The *OutputParSim* measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity, and a value of 0 represents a total difference.

Structural similarity is calculated as the average of parameter name similarity (Formula 1), input parameter similarity (Formula 2) and output parameter similarity

(Formula 5). Let $O_1 = (Oname_1, Ip_1, Op_1)$, $O_2, = (Oname_2, Ip_2, Op_2)$, be two Web service functions with their respective sets of parameters; the level of structural similarity between them is calculated as follows:

$$StructuralSim(O_1, O_2) = [\, FunctionNameSim\,(Oname_1, Oname_2) + InputParSim(O_1, O_2) + OutputParSim(O_1, O_2)\,] / 3 \quad (6)$$

The *StructuralSim* measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity, and a value of 0 represents a total difference.

4.1 Similarity Relations Discovery

The algorithm calculates *function* name similarity, input and output parameter similarities and structural similarities. If the resulting level of similarity is higher than an established threshold, then a similarity relationship is generated between both *functions* in the ontology. Thereafter, the ontology continues growing in similarity relationships as new services are registered. In this case, the resulting ontology is considered dynamic and evolving over time.

4.2 Inference Rules

In order to support updates in the ontology when new service instances are recorded, a set of rules for inference are specified.

The transitivity rule to derive new *Input Parameter* similarity instances is shown in Formula 7.

$$isInputParamSimilarTo(?a, ?b) \wedge isInputParamSimilarTo(?b, ?c) \rightarrow isInputParamSimilarTo(?a, ?c) \quad (7)$$

The transitivity rule to derive new *Output Parameter* similarity instances is shown in Formula 8.

$$isOutputParamSimilarTo(?a, ?b) \wedge isOutputParamSimilarTo(?b, ?c) \rightarrow isOutputParamSimilarTo(?a, ?c) \quad (8)$$

The transitivity rule to derive new *Function Name* similarity instances is shown in Formula 9.

$$isFunctionNameSimilarTo(?a, ?b) \wedge isFunctionNameSimilarTo(?b, ?c) \rightarrow isFunctionameSimilarTo(a, c) \quad (9)$$

The transitivity rule to derive new *Structural* similarity instances is shown in Formula 10.

$$isStructuralSimilarTo(?a, ?b) \wedge isStructuralSimilarTo(?b, ?c) \rightarrow isStructuralSimilarTo(?a, ?c) \quad (10)$$

Additional rules are specified to query the Ontology and obtain answers about the set of functions being treated. The query rule showed in Formula 11 displays pairs of Functions individuals for which a relation of *Input Parameter* similarity was discovered and established.

$$Functions(?x) \wedge Functions(?y) \wedge isInputParamSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (11)$$

Similarly, the query rule showed in Formula 12 displays pairs of Functions individuals for which a relation of *Output Parameter* similarity was discovered and established.

$$Functions(?x) \wedge Functions(?y) \wedge isOutputParamSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (12)$$

The query rule showed in Formula 13 displays pairs of Functions individuals for which a relation of *Function Name* similarity was discovered and established.

$$Functions(?x) \wedge Functions(?y) \wedge isFunctionNameSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (13)$$

Finally, the query rule presented in Formula 14 is very useful because it allows inferring what functions may be substitutable each other, provided they meet three conditions: *Input Parameter* similarity, *Output Parameter* similarity and *Function Name* similarity.

$$Functions(?x) \wedge Functions(?y) \wedge isInputParamSimilarTo(?x, ?y) \wedge isOutputParamSimilarTo(?x, ?y) \wedge isFunctionNameSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (14)$$

5 Experimental Setup

For experimentation 37 public Web service descriptions files were retrieved from *SeekdaTM*[5]. The architecture of this solution (see Fig. 2) consists of the following modules: a Web service *data extraction module*, which browses any set of public available WSDL files and extracts the service name, the set of function names, the names and data types of input and output parameters; an ontology *population module*, which registers into the ontology new function instances after data is extracted from WDSL files; and a *similarity relations discovery module*, which calculates structural similarities between function pairs and registers new semantic relations between compared individuals, if the level of similarity resulted higher than a threshold. The parser module extracts function names, input and output parameter names and types from 37 Web service description files. After the data extraction module is executed, the ontology is populated with a total of 537 new *Functions*, and 6317 *Parameters*: 3155 *InputParameters* and 3162 *OutputParameters*.

Table 1. Number of discovered relations for each measure

Similarity relationship	Total
Function name similarity	213
Input parameter similarity	470
Output parameter similarity	1440
Structural similarity	184

[5] http://webservices.seekda.com/

Discovered similarity relations are named *isFunctionNameSimilarTo*, *isOutputParamSimilarTo*, *isInputParamSimilarTo*, and *isStructuralSimilarTo*. For the set of 537 functions, similarity relationships were discovered between each function pair. Results are shown in Table 1.

The set of structural similar relationships is a combined result of the function name, input parameter and output parameter similarities for each function pair, which is the main reason of the reduced number of relations in comparison with the three previous.

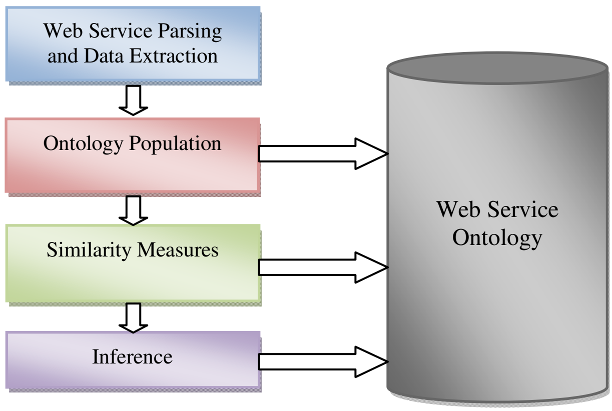


Fig. 2. Experimental architecture for automated similarity relations discovery

6 Conclusions

Results show advances on automatic similarity relations discovery from public available Web service descriptions. The automatic population of the ontology with existing WSDL files is a relevant advance towards the automated reutilization and construction of service-based solutions using pre-existing resources. Resulting similarities between service functions show that the set of measures calculations can be combined to obtain more complex and significant information concerning functions inter-relations.

The incorporation of transitivity rules for inference allows further discovery of more structural similarities between functions. This discovery process is executed over all functions, regardless of the service application domain; that is considering the possibility to find out similar functions from services belonging to different service types. In this paper, especial attention has been paid to structural similarities; but further similarities can be implemented to discover more information between functions. For instance, semantic distance metrics, linguistic patterns applied to discover similarities based on documentation provided in natural language, behavior measures, and data type comparison measures to find specific parameter set matches.

Inference is a key issue for maintainability and evolution of the ontology; inference rules generate new inter-relationships between functions and help to answer constrained queries regarding asserted inter-relationships in the ontology.

References

1. Dong, X., Halevy, A., Madhavan, J., Nemes, E., Zhang, J.: Similarity Search for Web Services. In: Proceedings of the 30th VLDB Conference, pp. 372–383 (2004)
2. Jianchun, F., Subbarao, K.: A Snapshot of Public Web Services. SIGMOD Rec. 34(1) (2005)
3. Rodríguez, M., Crasso, M., Zunino, A., Campo, M.: Improving Web Service descriptions for effective service discovery. Science of Computer Programming 75(11) (2010)
4. Rasmussen, E.: Clustering algorithms, Information retrieval: data structures and algorithms. Prentice-Hall, Inc., Upper Saddle River (1992)
5. Hamming, R.: Error detecting and error correcting codes. Bell System Technical Journal 29(2), 147–160 (1950)
6. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10, 707–710 (1966)
7. Plebani, P., Pernici, B.: URBE: Web Service Retrieval Based on Similarity Evaluation. IEEE Transactions on Knowledge and Data Engineering 21(11) (2009)
8. Elgazzar, K., Hassan, A.E., Martin, P.: Clustering WSDL Documents to Bootstrap the Discovery of Web Services (2010)
9. Bruno, M., Canfora, G., Di Penta, M., Scognamiglio, R.: An Approach to support Web Service Classification and Annotation. In: IEEE International Conference on e-Technology, e-Commerce and e-Service (2005)
10. Stroulia, E., Wang, Y.: Structural and Semantic Matching for Assessing Web Service Similarity. International Journal of Cooperative Information Systems 14(4) (2005)
11. McIlarith, S., Cao Son, T., Zeng, H.: Semantic Web Services. IEEE Intelligent Systems (2001)
12. Bener, A., Ozadali, V., Ilhan, E.S.: Semantic matchmaker with precondition and effect matching using SWRL. International Journal of Expert Systems with Applications, 9371–9377 (2009)
13. Klusch, M., Kaufer, F.W.-M.: A Hybrid Semantic Web Service Matchmaker. Web Intelligence and Agent Systems 7(1), 23–42 (2009)
14. Jeong, B., Cho, H., Lee, C.: On the functional quality of service (FQoS) to discover and compose interoperable Web services. International Journal of Expert Systems with Applications, 5411–5418 (2008)
15. Grigori, D., Corrales, J., Bouzeghoub, M.: Behavioral matchmaking for service retrieval. In: IEEE International Conference on Web Services (2006)
16. Dumas, M., García-Bañuelos, L., Dijkman, R.: Similarity Search of Business Process Models. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 32(3), 23–28 (2009)
17. Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S., Zave, P.: Matching and Merging of Statecharts Specifictaions. In: Proceedings of the 29th International Conference on Software Engineering (2007)
18. Gruber, T.R.: A Transaltion approach to portable ontologies. Knowledge Acquisition 5(2), 199–220 (1993)
19. Bravo, M., Alvarado, M.: Similarity Measures for Substituting Web Services. International Journal of Web Services Research 7(3) (2010)
20. Gupta, C., Bhowmik, R., Govindaraju, M.: Web Services Operation and Parameter Matchmaking Based on Free-Form User Queries. In: Proceedings of Congress on Services Part II, pp. 57–64 (2008)

The Role of Class Dependencies in Designing Ontology-Based Databases

Chedlia Chakroun, Ladjel Bellatreche, and Yamine Ait-Ameur

LISI/ENSMA - Poitiers University Futuroscope, France
{chakrouc,bellatreche,yamine}@ensma.fr

Abstract. Recently, an important number of applications are producing and manipulating mountains of ontological data. Managing them efficiently needs the development of scalable solutions. Ontology-based databases (OBDB) are one of these solutions. An OBDB stores both ontological data and the ontology describing their meanings in the same repository. Several architectures supporting these OBDB were proposed by academicians and industrial editors of DBMS. Unfortunately, there is no available methodology for designing such OBDB. To overcome this limitation, this paper proposes to scale up the traditional database design approaches to OBDB. Our approach covers both conceptual and logical modeling phases. It assumes the availability of a domain ontology composed by primitive (canonical) and defined (non-canonical) concepts. Dependencies among properties and classes are captured and exploited to define a normalized logical model. A prototype implementing our design methodology on the OBDB OntoDB is outlined.

1 Introduction

Roughly speaking, ontologies have been introduced in information systems as knowledge models that provide with definitions and descriptions of the concepts of a target application domain [14,16,18,19]. Therefore, the number of ontological data manipulated and produced by different applications in various domains has increased. This is mainly due to three factors: (i) the development of domain ontologies, (ii) the availability of software tools for building, editing and deploying ontologies and (iii) the existence of ontology model formalisms (OWL, PLIB, etc.) which have considerably contributed to the emergence of ontology based applications. Facing this situation, managing, storing, querying these data and reasoning on them require the development of software tools capable of handling these activities. Moreover, due to the wide usage of such data, the need of a systematic and consistent development process appeared, taking into account the central objectives of persistence, scalability and high performance of the applications. Persistence of ontological data [7] was advocated by academic and industrial researchers. As a consequence, a new type of databases, named *ontology-based databases* (OBDB) dedicated to store and manage the emerged ontological data. An OBDB is a database model that allows both ontologies and their instances to be stored in a single repository [9]. Since an OBDB is a

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 444–453, 2011.
© Springer-Verlag Berlin Heidelberg 2011

database, it should be designed according to the classical design process dedicated to the database development, identified in the ANSI/X3/SPARC architecture. At the origin, this architecture does not recommend the use of ontologies as a domain knowledge model, but it refers to the notion of dictionary. In [10], an extension of this architecture giving ontologies their right place during the life cycle of database scheme process design has been proposed. But, the reality is different. Indeed, when exploring the literature, we figure out that most of the research efforts were concentrated on the physical design phase, where various storage models for ontological data were given. OntoDB [9], Sesame [4], Oracle [8], DLDB2[15], etc. are examples of such systems. Note that each system is designed according to a given architecture and has its own storage model. Three main architectures have been proposed. In this first architecture, the ontology and its associated data are stored in an RDF triples structure: *(subject, predicate, object)*. There is no identified separation between ontology and data. This architecture is proposed in Oracle [8], etc. In the second architecture, ontology and its ontological data are stored independently into *two different schemes*. Sesame is an example of such architecture. The storage model used for the ontology is based on RDFS, whereas data may be represented using different storage models. These two architectures share a common property: the ontology model (RDF or RDFS) is fixed. Unlike logical models in the ANSI/X3/SPARC architecture, their physical structure is static and does not evolve according to the stored ontological data model. The third architecture extends the second one, by adding a new part, called, the *meta-schema part*. This part offers flexibility of the ontology part modeling, since it is represented as an instance of the meta-schema. Indeed, it may allow: (1) a generic access to the ontology part, (2) the evolution of the used ontology model by adding non-functional properties (functional dependencies [2], Web services, etc.) and (3) a storage of different ontology models (OWL, PLIB, etc.). This third architecture is proposed in OntoDB [9], with PLIB as the ontology model. For instance, OntoDB [9] uses a *horizontal storage* model, where a single table is associated to each ontology class with one column per each property. Unlike previous architectures where the ontology model is fixed, the third architecture supports the evolution of the ontology model through the meta-schema. To facilitate the exploitation of these OBDB systems, different query languages were proposed (OntoQL [12], SPARQL[1], etc.). Based on the spectacular development of OBDB, the development of an OBDB design methodology becomes a crucial issue for companies. In order to encourage designers to advocate a such methodology, it should then *scale up* the traditional database design approaches by offering similar steps: *conceptual*, *logical* and *physical*. Actually, to design an OBDB from a given domain ontology, we have to choose a favourite architecture, to identify the relevant storage models and to establish mappings between ontological concepts and the target entities of the chosen storage model. This design procedure suffers from (a) the presence of duplicated and inconsistency data when populating the target OBDB (lack of integrity constraints) and (b) the lack for *data access transparency*.

[1] http://www.w3.org/TR/rdf-sparql-query/

Recently, some research studies recommending the ontologies use to design traditional databases and data warehouses have been described in the literature. The similarities between ontologies and conceptual models [18] and the *reasoning* capabilities offered by ontologies have motivated such studies. Indeed, [19] used linguistic ontologies to design traditional databases while [14,16] used conceptual ontologies to design multidimensional data warehouses. We may observe that in the last decade, data warehouses faced the same phenomena as for OBDBs, where most of the studies focused on the physical design [6]. So, although conceptual design and requirement analysis are two key steps within the database and data warehouse design processes [11], they were neglected in the first era of OBDBs.

In classical databases, redundancy is addressed by normalizing the logical model obtained from a conceptual model. This normalization process is performed thanks to the exploitation of the available functional dependencies (FD) between properties. Recently, a couple of studies enriched ontology models by FD defined on properties [2,5]. In [2], we proposed a design methodology for OBDB by considering FD defined only on properties of canonical classes. In the context of ontologies, two types of classes may be distinguished: (1) canonical (CC) and (2) non-canonical (NCC) (see Section 2.1). The presence of the non-canonical classes generates a new form of dependencies between classes [17].

The objectives of our work consist in (1) exploiting definitions of classes to extract class dependencies (CD), (2) exploiting CD to identify CC and NCC, (3) developing a methodology for designing OBDB, including the conceptual and logical models in order to identify the persistent classes (CC) and the views (NCC) and (4) deploying the methodology on a particular OBDB system.

This paper is divided into five sections. Section 2 describes the basic concepts related to dependencies between ontological concepts, a formal ontology model and the notion of dependency graph. The main steps of our methodology are presented in Section 3. Section 4 shows a validation of our methodology on the LUB ontology. Finally, Section 5 concludes the paper by summarizing the main results and suggesting future work.

2 Basic Concepts and Formalization

In this section, we first present some concepts, definitions and formalizations related to (i) ontologies and (ii) dependency graph exploiting class dependencies.

2.1 Taxonomy of Ontologies

A taxonomy of ontologies has been proposed in [13] where three layers are identified: (1) *Conceptual Canonical Ontologies (CCO)* can be considered as shared conceptual models containing the core classes concepts, (2) *Non Conceptual Canonical Ontologies* extend CCO by allowing the designers to introduce derived classes and concepts and (3) *Linguistic Ontologies*. This taxonomy helps designers to identify CC and NCC of an ontology as shown in the next example.

Example 1. Figure 1 represents an *extended fragment* of the *Lehigh University Benchmark (LUB)*[2], where the *University (U)* class defined as the union of the two CC: *PublicUniversity (PU)* and *PrivateUniversity (PRU)* ($U \equiv PU \cup PRU$) is added. Based on the ontology definition, CC and NCC are identified as follows: $NCC^{LUB} = \{U, Student\_Employee, MasterCourse, FrenchPrivateUniversity\}$, $CC^{LUB} = C^{LUB} - NCC^{LUB}$, where C^{LUB} represents all LUB classes. Considering that NCC are defined from CC, a dependency relation between CC and NCC may exist. For example, if we consider the definition of U, a dependency is defined as: $(PU, PRU) \longmapsto U$. Note that other types of dependencies between ontological concepts (see the next section) have been studied in the literature.

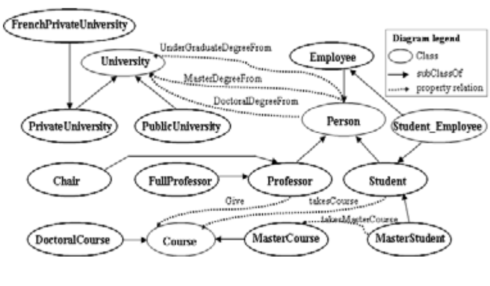


Fig. 1. Extended LUB Ontology

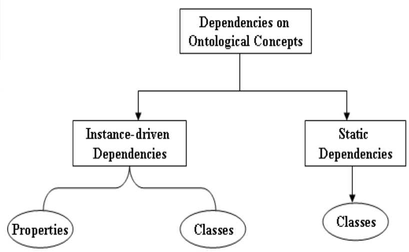


Fig. 2. Dependencies Classification

2.2 Dependencies between Ontological Concepts

In this section, we show the interest of capturing dependencies between ontological concepts for designing OBDB. Our dependencies analysis [2,5,17] gives rise to the following classification (Figure 2): *instance driven dependencies* (IDD) and *static dependencies* (SD). IDD is quite similar to functional dependencies (FD) in traditional databases. IDD may concern either properties [5] (IDD^P) or classes [17] (IDD^C) of the ontology. In [5], authors proposed a formal framework for handling FD constructors for any type of OWL ontology. Three categories of FDs are identified. In [2], we supposed the existence of FD involving simple properties of each ontology class. For instance, if we consider a class *Student* with a set of properties *id* and *age*, the FD $id \rightarrow age$ may be defined. In [17], the authors proposed an algorithm to discover FD among concepts of an ontology that exploits the inference capabilities of DL-Lite. A FD among two concepts C_1 and C_2 ($C_1 \rightarrow C_2$) exists if each instance of C_1 determines one and only one instance of C_2. For instance, if we consider a role *mastersDegreeFrom* with a domain and a range *Person* and *University*, the FD $Person \rightarrow University$ is defined. SD are defined between classes based on their definitions (Example 1). A SD between two concepts C_i and C_j ($C_i \longmapsto C_j$) exists if the definition of

[2] The ontology and its class definitions described in OWL are available at: `http://www.lehigh.edu/ zhp2/2004/0401/univ-bench.owl`

C_i is available then C_j can be derived. This definition is supported by a set of OWL[3] constructors. For example, if we consider a class *Student* having *level* as one of its properties, a class *MasterStudent* may be defined as a *Student* with a master level and the dependency $Student \longmapsto MasterStudent$ is obtained.

2.3 A Formal Model for Ontologies

An ontology $\mathcal{O}$ may be defined as $< \mathcal{C}, \mathcal{P}, Applic, Sub, PFD, CD >$, where:

- $\mathcal{C}$ is the set of the classes used to describe the concepts of a given domain.
- $\mathcal{P}$ is the set of all properties used to describe the instances of $\mathcal{C}$.
- *Applic* is a function defined as $Applic : \mathcal{C} \rightarrow 2^{\mathcal{P}}$. It associates to each class of $\mathcal{O}$, the properties that are applicable for each instance of this class.
- *Sub* is the subsumption relationship defined as $Sub : \mathcal{C} \rightarrow 2^{C}$, where for a class $C_i \in \mathcal{C}$, it associates its direct subsumed classes[4].
- PFD: a mapping from the powerset of $\mathcal{P}$ onto $\mathcal{P}$ ($2^{\mathcal{P}} \rightarrow \mathcal{P}$) representing IDD defined on the applicable properties of $\mathcal{P}$ of a class $C_i \in \mathcal{C}$ (IDD^{P}).
- CD: a mapping from the powerset of $\mathcal{C}$ onto $\mathcal{C}$ ($2^{\mathcal{C}} \rightarrow \mathcal{C}$) representing either static dependencies (SD) or instance-driven dependencies defined on classes (IDD^{C}). $CD = SD \cup IDD^{C}$, where:
  - IDD^{C}. Let I_1, I_2 be respectively the population of C_1 and C_2. IDD^{C}: $C_1 \rightarrow C_2$ exists if for each $i_k \in I_1$, there exists a unique $i_j \in I_2$ such that i_k determines i_j.
  - SD . Let I_i, I_j be respectively the population of C_i and C_j. $SD : C_i \longmapsto C_j$ means that I_i determines I_j.

To facilitate the representation of dependencies between classes, we adopt a graphical representation discussed in the next section.

2.4 A Formal Model for Dependency Graph

Dependencies between different ontology classes may be represented by a directed graph $\mathcal{G}$, called a *dependency graph*. Formally, $\mathcal{G}$ may be defined as a pair (C^{LO}, A), where C^{LO} are the nodes and A are edges. An edge $a_k \in A$ between a pair of classes C_i and C_j ($\in C^{LO}$) exists, if a CD between C_i and C_j is established i.e. $C_i \longmapsto C_j \in SD$. Figure 3 presents an example of a dependency graph of the LUB ontology. A minimum coverage-like classes $\mathcal{C}^{+}$ may be deduced from $\mathcal{G}$ as a minimal set of CD allowing the knowledge of the set of all CD. Therefore, $\mathcal{C}^{+}$ is a subset of CD. Consider the dependency graph illustrated in Figure 3. The minimum coverage-like classes is $\mathcal{C}^{+} = \{C_2, C_3 \longmapsto C_1; C_1 \longmapsto C_3; C_5, C_6 \longmapsto C_4; C_9, C_8 \longmapsto C_7; C_3 \longmapsto C_{15}\}$.

3 Our Proposal

In this section, we propose a complete methodology to design OBDB from an OWL domain ontology $\mathcal{O}$ respecting the ontology formal model.

[3] `http://www.w3.org/TR/owl-guide/`

[4] C_1 subsumes C_2 iff $\forall x$ instance of C_2, x is instance of C_1.

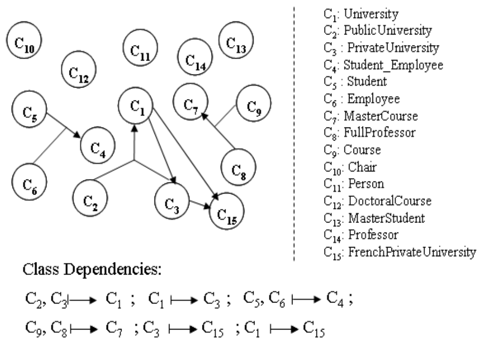


Fig. 3. Example of a dependency graph

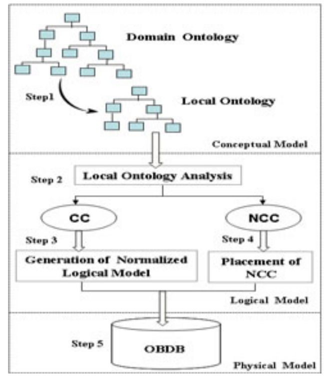


Fig. 4. Steps of our approach

3.1 Different Phases of our Methodology

Inspired from the database design process, our methodology starts from a conceptual model to provide logical and physical models. It is a five step method (Figure 4). It starts from the extraction of a local ontology (step 1) and then identifies CC and NCC by exploiting CD (step 2). As a further step, it defines in parallel a placement of the NCC (step 4) and a logical model for each CC (step 3). Finally, a logical model for the OBDB is generated (step 5).

Step 1. The designer extracts a fragment of $\mathcal{O}$ (called local ontology(LO)) according to her/his requirements. The LO plays the role of conceptual model. Three extraction scenarios may occur: (1) $LO = \mathcal{O}$ means that $\mathcal{O}$ covers all the designer requirements. (2) $LO \subset \mathcal{O}$ means that $\mathcal{O}$ is rich enough to cover the user requirements. (3) $LO \supseteq \mathcal{O}$ means that $\mathcal{O}$ does not fulfill the whole designer requirements. The designer extracts from the $\mathcal{O}$ a fragment corresponding to the requirements and enriches it by adding new concepts/properties/dependencies.

Step 2. Once the LO is extracted, an analysis is required to identify CC and NCC. Let C^{LO}, CC^{LO} and NCC^{LO} be the set of all, canonical and non-canonical classes of the LO. A dependency graph is built from CD. This graph is used to determine the minimum set of CC. Our graph dependency is quite similar to the FD graph defined for classical databases to generate a minimum coverage and normalized tables [1]. The difference is that we have classes as nodes whereas attributes represent nodes in a FD graph. Based on this similarity, we adapt [1]'s algorithm to generate CC^{LO} (for lack of space this algorithm is described in technical report available at: http://www.lisi.ensma.fr/ftp/pub/documents/reports/2011/2011-LISI-3.pdf). This algorithm has as input CD and generates CC. It starts by the calculation of isolated classes ($C_{Isolated}$: classes not involved in

CD). These classes have to be canonical since they can not be derived from other ones. Then, the minimum coverage-like classes ($\mathcal{C}^+$) is computed. It represents the minimum subset of basic CD to generate all the others. $\mathcal{C}^+$ has to be treated so as to eliminate cycle relationships between classes. Then, CC are deduced. For example, if we consider the dependency graph described in Figure 3, $C_{Isolated}$ are $\{C_{10}, C_{11}, C_{12}, C_{13}, C_{14}\}$. Then, a cycle between C_1, C_2 and C_3 is identified ($C_1 \longmapsto C_3; C_2, C_3 \longmapsto C_1$). Therefore, the CD: $C_1 \longmapsto C_3$ is eliminated. Finally, CC^{LO} are calculated as $\{C_2, C_3, C_5, C_6, C_8, C_9, C_{10}, C_{11}, C_{12}, C_{13}, C_{14}\}$. Note that different sets of CC may be obtained. In this case, the designer may choose his relevant set. This distinction between class types is important in order to optimize the data representation and reduce redundancy in the final OBDB.

Step 3. Based on the CC^{LO} and NCC^{LO}, two scenarios are distinguished:

1- $NCC^{LO} = \phi$: Only CC^{LO} exists. Then, the FD defined on their properties are used for normalization and for the definition of their primary keys [2].

2- $CC^{LO} \neq \phi$ and $NCC^{LO} \neq \phi$. For each class in CC^{LO}, the same mechanism described in Step 3.1 is applied. Then, for each class in NCC^{LO}, a relational view is computed. For example, let $ncc_j \in NCC^{LO}$ be a NCC defined as the union of two CC $cc_1(p_1, p_2, p_3)$ and $cc_2(p_1, p_2, p_4)$, a view corresponding to ncc_j is defined as follows: $((Select\, p_1, p_2\ From\ cc_1)\ Union\ (Select\, p_1, p_2\ From\ cc_2))$. One of the advantages of using views to represent NCC is to ensure the transparency in accessing data. A user may query an OBDB via these classes without worrying about the physical implementation of those classes.

Step 4. The C^{LO} may be defined without specifying the subsumption relationship. Thus, we propose to store all C^{LO} regardless of their types (CC and NCC) in the OBDB taking into account the subsumption hierarchy of classes using a reasoner such as Pellet [3], etc. For example, the classes PU and PRU ($U \equiv PU \cup PRU$), will be subclasses of the $University$ class.

Step 5. Now, the database administrator may choose any existing database architecture offering the storage of ontology and ontological data.

4 Case Study

In this section, we propose a case study of our design approach. We use OntoDB [9] as storage model architecture for our *physical design phase* (Step 5) for two main reasons: (i) it belongs to the third architecture (Section 1) that allows us to enrich the meta-schema by dependencies. (ii) A prototype is available, it has been used in several industrial projects and it is associated with a query language called *OntoQL* [12] (SPARQL like language) defined as an extension of SQL. It offers DDL statements for creating entities, classes and views. Our two algorithms for determining CC and NCC and for normalizing each class using PFD were encoded within Java language.

To validate our proposal, a deployment process is done in OntoDB onto three parts: meta-schema, ontology and data.

4.1 Meta-Schema Part Deployment

In this section, we show how dependencies and canonicity identification can be made persistent in OntoDB. The meta-schema of OntoDB contains two main tables *Entity* and *Attribute* encoding the meta-meta-model level of the MOF architecture. To conduct our validation, the support of both ontological dependencies and canonicity in the *initial kernel* of OntoDB is required. To do so, we extended the *meta-schema* of OntoDB by incorporating dependencies and canonicity concepts. First, we develop a meta-model describing both dependencies and concepts type (CC and NCC). Figure 5.(a) shows the UML model of this meta-model (MMPCD). The following OntoQL statement instantiate the tables *Entity* and *Attribute* (Figure 5.(b)) with class dependency description.

```
CREATE ENTITY #CD.LP (#its.LP.Classes REF(#Class) ARRAY)
CREATE ENTITY #CD.RP (#its.RP.Class REF(#Class) )
CREATE ENTITY #CD (#its.CD.RP REF(#CD.RP), #its.CD.LP REF(#CD.LP))
```

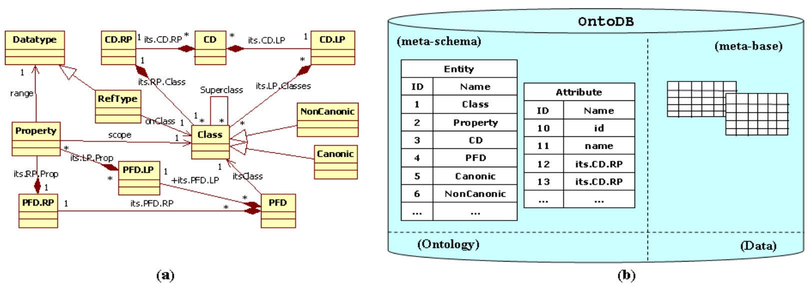


Fig. 5. (a) MMPCD of meta-model, (b) Extended OntoDB by PFDs and CDs

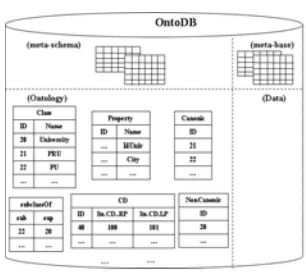


Fig. 6. Ontology part Deployment

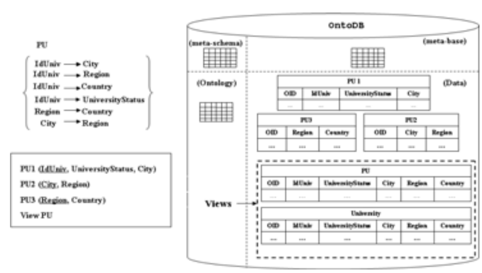


Fig. 7. Data part Deployment in OntoDB

Once this model is encoded, the obtained extended meta-schema gives us the possibility to store *ontologies with their dependencies* and class types.

4.2 Ontology Part Deployment

Once the meta-schema is extended, the LO is created. After creating the structure of a class, their dependencies should be attached. For example, the dependency $U \longmapsto PU$ is defined by the following OntoQL statement:

```
INSERT INTO #CD (#its.CD.RP, #its.CD.LP) VALUES((SELECT #oid from #Class c
WHERE c.#name='PU'),(SELECT #oid from #Class c WHERE c.#name='University'))
```

Instantiation of dependencies between properties is handled in a similar way. To identify CC and NCC (Step 2), we use the algorithm described in Section 3.1. To assure the subsumption relationships (Step 4), classes are placed in the ontology hierarchy of OntoDB (*subclassOf* Table) using Pellet 1.5.2 [3]. Figure 6 presents deployment results in the ontology part.

4.3 Data Part Deployment

In this part, we store the generated normalized logical model related to the used ontology (Step 3). We exploit PFD defined on each CC to generate normalized relations per CC. For NCC, views are generated. For instance, a view corresponding to the NCC University ($U \equiv PU \cup PRU$) is generated as follows:

```
CREATE VIEW University as ((SELECT*FROM PRU) UNION (SELECT*FROM PU))
```

Figure 7 shows an example of the generated normalized logical model of the CC $PublicUniversity$ and the NCC $University$. Up-to-date, the proposed methodology has been validated by a deployment on a third type architecture with OntoDB. However, it can be deployed on any database system if its ontology model supports dependencies between ontological concepts. Our approach generates a normalized logical model from a given domain ontology. Once the logical modeling phase is achieved, several scripts in different query languages (SQL, OntoQL, XML, etc.) are generated and run on the appropriate database.

5 Conclusion

This paper presented a five step methodology handling the consistent design of an OBDB from the conceptual model till the logical model. The approach considers LOs as conceptual models and borrows formal techniques issued from graph theory for dependency analysis, from description logics reasoning for class placement and from relational database theory for creating relational views. To the best of our knowledge, this work is the sole that considers different dependencies types between ontological concepts at different design levels of an integrated methodology. The different refinement of our approach preserve the functionalities offered by the LO. It is based on formal models and is independent of the chosen OBDB architecture. A deployment of our methodology in a particular OBDB system (OntoDB) is proposed. Currently, we are studying the problem of the deployment of OBDB based on different database architectures and storage models in order to cover the different phases of the life cycle of OBDBs.

References

1. Ausiello, G., D'Atri, A., Saccà, D.: Graph algorithms for functional dependency manipulation. Journal of the ACM 30(4), 752–766 (1983)
2. Bellatreche, L., Aït Ameur, Y., Chakroun, C.: A design methodology of ontology based database applications. Logic Journal of the IGPL (2010)
3. Evren, S., Bijan, P.: Pellet: An owl dl reasoner. In: International Workshop on Description Logics (DL 2004), pp. 6–8 (2004)
4. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
5. Calbimonte, J.P., Porto, F., Maria Keet, C.: Functional dependencies in owl abox. In: Brazilian Symposium on Databases (SBBD), pp. 16–30 (2009)
6. Chaudhuri, S., Narasayya, V.: Self-tuning database systems: A decade of progress. In: VLDB, pp. 3–14 (September 2007)
7. Chen, L., Martone, M.E., Gupta, A., Fong, L., Wong-Barnum, M.: Ontoquest: Exploring ontological data made easy. In: Proceedings of the International Conference on Very Large Databases, pp. 1183–1186 (2006)
8. Das, S., Chong, E.I., Eadon, G., Srinivasan, J.: Supporting ontology-based semantic matching in rdbms. In: VLDB, pp. 1054–1065 (2004)
9. 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)
10. Fankam, C., Jean, S., Bellatreche, L., Aït-ameur, Y.: Extending the ANSI/SPARC Architecture Database with Explicit Data Semantics: An Ontology-based Approach. In: Morrison, R., Balasubramaniam, D., Falkner, K. (eds.) ECSA 2008. LNCS, vol. 5292, pp. 318–321. Springer, Heidelberg (2008)
11. Golfarelli, M., Rizzi, S.: Data Warehouse Design: Modern Principles and Methodologies. McGraw Hill (2009)
12. Jean, S., Aït Ameur, Y., Pierra, G.: Querying ontology based databases - the ontoql proposal. In: Proceedings of the 18th International Conference on Software Engineering & Knowledge Engineering (SEKE 2006), pp. 166–171 (2006)
13. Jean, S., Pierra, G., Aït Ameur, Y.: Domain ontologies: A database-oriented analysis. In: WEBIST (1), pp. 341–351 (2006)
14. Nebot, V., Berlanga, R., Pérez, J.M., Aramburu, M.J., Pedersen, T.B.: Multidimensional Integrated Ontologies: A Framework for Designing Semantic Data Warehouses. In: Spaccapietra, S., Zimányi, E., Song, I.-Y. (eds.) Journal on Data Semantics XIII. LNCS, vol. 5530, pp. 1–36. Springer, Heidelberg (2009)
15. Pan, Z., Zhang, X., Heflin, J.: Dldb2: A scalable multi-perspective semantic web repository. In: International Conference on Web Intelligence, pp. 489–495 (2008)
16. Romero, O., Abelló, A.: Automating multidimensional design from ontologies. In: DOLAP, pp. 1–8 (2007)
17. Romero, O., Calvanese, D., Abelló, A., Rodriguez-Muro, M.: Discovering functional dependencies for multidimensional design. In: DOLAP, pp. 1–8 (2009)
18. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. SIGMOD Record 31(4), 12–17 (2002)
19. Sugumaran, V., Storey, V.C.: The role of domain ontologies in database design: An ontology management and conceptual modeling environment. ACM Transactions on Database Systems 31(3), 1064–1094 (2006)

An Access Control Model for Linked Data*

Serena Villata, Nicolas Delaforge, Fabien Gandon, and Amelie Gyrard

INRIA Sophia Antipolis
firstname.lastname@inria.fr

Abstract. Linked Open Data refers to a set of best practices for the publication and interlinking of structured data on the Web in order to create a global interconnected data space called *Web of Data*. To ensure the resources featured in a dataset are richly described and, at the same time, protected against malicious users, we need to specify the conditions under which a dataset is accessible. Being able to specify access terms should also encourage data providers to publish their data. We introduce a lightweight vocabulary, called Social Semantic SPARQL Security for Access Control Ontology (S4AC), allowing the definition of fine-grained access control policies formalized in SPARQL, and enforced when querying Linked Data. In particular, we define an access control model providing the users with means to define policies for restricting the access to specific RDF data, based on social *tags*, and *contextual* information.

Keywords: LOD, Security, SPARQL 1.1, Context, Named Graphs.

1 Introduction

Linked Data[1] [6] enables us to set links between items in different data sources, and to connect these sources into a single global data space. These data are provided with machine-readable annotations called *metadata*. Metadata have the aim to provide a flexible way to describe things, and how they relate to other things. However, one of the challenges of Linked Data is access control. As underlined by Bizer et al. [2,6], the datasets are published in the Linked Open Data (LOD) cloud[2] without the addition of any kind of metadata specifying the access control conditions under which the data is accessible.

This paper addresses the research question: *How to define an access control model for Linked Data?* This is important in order to encourage as many data providers as possible to publish data in their own terms, and not only fully public data. The research question breaks down into two sub-questions: (i) *how to define fine-grained access policies?* and (ii) *how to define context-based access policies?*

The issue of defining access control policies for the Web has been addressed by the Web Access Control vocabulary (WAC)[3], which allows the user to specify

* DataLift is funded by the French National Research Agency: ANR-10-CORD-09.

1 http://www.w3.org/DesignIssues/LinkedData.html

2 http://richard.cyganiak.de/2007/10/lod/

3 http://www.w3.org/wiki/WebAccessControl

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 454–463, 2011.
© Springer-Verlag Berlin Heidelberg 2011

access control lists (ACL). The ACL are of the form [acl:accessTo <card.rdf>; acl:mode acl:Read, acl:Write; acl:agentClass <groups/fam#group>], which means that anyone in the group <http://example.net/groups/fam#group> may read and write card.rdf. The WAC vocabulary distinguishes four classes of access control privileges: Read (read the content), Write (delete or update the content), Control (set the ACL for the content), and Append (add information at the end of the content). This vocabulary grants the access to a whole RDF document, e.g., card.rdf. In this paper, we aim at providing fine-grained access control policies which grant the access to specific RDF data, i.e., the information providers may want to restrict the access to a few named graphs [4]. Moreover, we enable the requester to submit any SPARQL query, and resource provider to further specify the access control privilege granted to the user, and we distinguish the Delete and Update classes of privileges, included in the Write WAC class.

We introduce the Social Semantic SPARQL Security for Access Control vocabulary (S4AC), a lightweight ontology which allows the information providers to specify fine-grained access control policies for their RDF data (Figure 1). At the core of S4AC is the Access Condition which is a SPARQL 1.1. ASK clause that specifies the condition to be satisfied in order to grant the access to a named graph. Moreover, the information providers can define Access Conditions based on *tags* which restrain the conditions to named graphs tagged with such tags, e.g., named graphs tagged "friends", "amici", "ami". The conditions can be bound on specific values to provide an access evaluation context, e.g., <''?user'', <http://myExample.net#sery>> where the URI of the user is bound to <http://myExample.net#sery>. Finally, the Access Condition is associated with a temporal validity. The Access Privilege defines which kind of privilege is granted to the user satisfying the Access Conditions, e.g., s4ac:Update grants the user the privilege to modify the requested named graph.

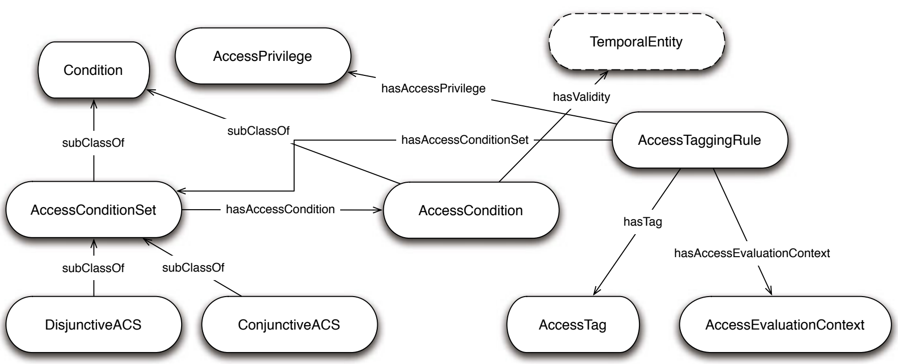


Fig. 1. An overview of the S4AC Ontology

A key feature of our approach is to rely only on Semantic Web languages. As a consequence, our access control model is platform independent, and can be used by any kind of system based on those languages. In particular, the semantics of our access control policies is grounded in SPARQL 1.1[4] `ASK` queries. Relying on SPARQL semantics, our model allows the user to submit arbitrary queries while enforcing fine-grained access rules on the results he will receive. If the result of the `ASK` query is *true*, then the user is provided with the information he requires. If the result is *false*, then the model returns to the user a denial coupled with one or more rule labels explaining the denial.

The reminder of the paper is organized as follows: Section 2 presents a use case of the proposed access control model. Section 3 introduces the S4AC ontology, and it details and analyses the access control polices which can be defined using our model. Related work and conclusions end the paper.

2 The DataLift Use Case

The DataLift project[5] aims at providing a platform to ease the publication and interlinking of datasets on the *Web of Data*. Figure 2 illustrates the Access Control Manager (ACM) which is the core module of our access control model. We now describe the features of the Access Control Manager first from the point of view of the user, and then from the point of view of the system.

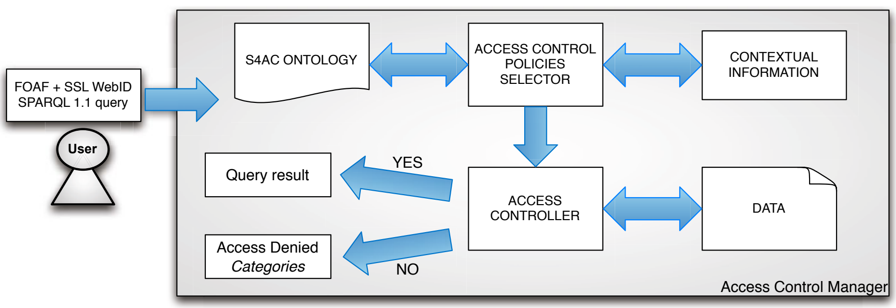


Fig. 2. The Access Control Manager

Consider a user who wants to access some of the information published on the Web of Data by means of the DataLift platform. The user first authenticates to the ACM of the platform using the WebID protocol[6]. The user queries the datasets using a SPARQL 1.1 `SELECT`, `MODIFY`, `INSERT`, or `DELETE` query[7], depending on the kind of operation the user intends to perform on the requested

[4] `http://www.w3.org/TR/sparql11-query/`
[5] `http://datalift.org/en/`
[6] `http://www.w3.org/wiki/WebID`
[7] The `MODIFY`, `INSERT`, and `DELETE` queries are provided by SPARQL 1.1. See `http://www.w3.org/TR/2009/WD-sparql11-update-20091022/`.

data. The ACM returns the user an answer of the kind YES/NO together with the query result, or the labels of the rules that caused the failure.

The ACM receives an authentication request from the user by means of the WebID protocol. Then, after a successful authentication, it receives the query of the user. The ACM has the aim to grant or restrict the access to the RDF data published using the DataLift platform, where each SPARQL endpoint manages its requests. Once the request of the user is received, the ACM selects, by means of the module called Access Control Policies Selector (ACPS), which policy applies, depending on the requested operation. For instance, if the user submitted a `MODIFY` query, then the ACPS identifies all the policies which apply, and concern an `Update` access privilege. The ACPS handles two kinds of operations: (i) it checks the S4AC ontology in order to identify which access conditions apply, and (ii) it checks whether the contextual information, e.g., the temporal validity of the selected policies, is satisfied. Note that we check whether the contextual constraints hold before checking the reminder of the policy. If the contextual constraints are not satisfied, we already know that the access will not be granted. After the identification of the policies, and a positive checking of the contextual constraints, the Access Controller module matches the policies according to the user's profile to test what he can access. The Access Controller addresses a SPARQL `ASK` query which returns *true* if the access to the named graph is granted to the user. The Access Controller selects the set of named graphs to which the user has access, and queries this dataset adding `FROM, FROM NAMED` to the user's query. If the answer is *false*, then the Access Controller returns a failure, coupled with the categories causing the failure. These categories are provided to the Access Controller by the ACPS when it checks the ontology.

3 Access Control for Linked Data

3.1 Social Semantic SPARQL Security for Access Control Ontology

The Social Semantic SPARQL Security for Access Control Ontology (S4AC), online at `http://ns.inria.fr/s4ac/v1#`, is detailed in Figure 3. One of the key features of our access control approach is to be integrated with the models adopted in the fields of the Social Web, and of the Web of Data. In particular, S4AC reuses concepts from SIOC[8], SCOT[9], NiceTag[10], WAC, TIME[11], and the access control model as a whole is grounded on further existing ontologies, as FOAF[12], Dublin Core[13], and RELATIONSHIPS[14].

The main class of the S4AC ontology is the class *AccessCondition*, which is a subclass of the class *Condition*, itself a subclass of `sioc:Item`.

[8] `http://rdfs.org/sioc/spec/`
[9] `http://scot-project.net/`
[10] `http://ns.inria.fr/nicetag/2010/09/09/voc.html`
[11] `http://www.w3.org/TR/2006/WD-owl-time-20060927/`
[12] `http://xmlns.com/foaf/spec/`
[13] `http://dublincore.org/documents/dcmi-terms/`
[14] `http://vocab.org/relationship/.html`

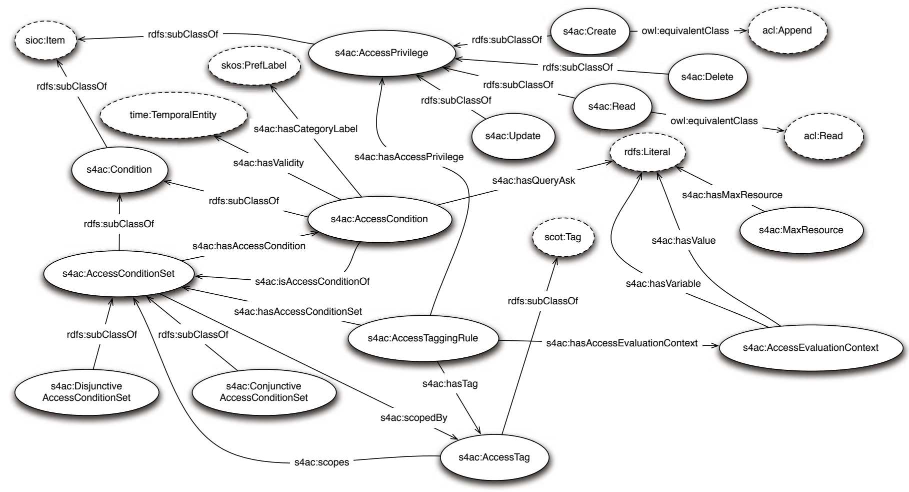


Fig. 3. The S4AC Ontology

Definition 1. *An Access Condition (AC) is a SPARQL 1.1* `ASK` *query. If a solution exists, the* `ASK` *query returns true, and the Access Condition is said to be verified. If no solution exists the* `ASK` *query returns false, and the Access Condition is said not to be verified.*

The *Access Condition* grants or restricts the access to the data. If the `ASK` returns *true*, the access is granted to the user. In order to return the user a more informative answer if the access is denied, we introduce the property *hasCategoryLabel*. This property allows to associate to each *AC* one or more natural language labels which "identify" the access condition, and they are returned to the user to provide him the reasons of the denial. We cannot return the user all the access conditions, because this would make him aware of the policies of the provider. The *AccessCondition* defines the property of the access polices *hasValidity*. It allows to define the validity of an Access Condition. Thanks to the use of the concept `time:TemporalEntity`, the validity can be expressed in various ways: valid from/through a specific date/time, or valid in a specific interval of time. This property is used to express policies in which not only the identity of the user requesting the data is checked, but also the contextual information related to the time in which the request is performed. A further class is *MaxResource* which defines the number of times the user can access all or a specified named graph. This class has the property *maxOnResource* which is used to precise which resource is accessible by a limited number of accesses.

Definition 2. *An Access Evaluation Context (AEC) is a list L of predetermined bound variables of the form* $L = (\langle var_1, val_1\rangle, \langle var_2, val_2\rangle, \ldots, \langle var_n, val_n\rangle)$ *that is turned into a SPARQL 1.1 Binding Clause to constrain the* `ASK` *query evaluation when verifying the Access Conditions.*

The *AEC* is represented in the ontology as the class *AccessEvaluationContext* which has two properties, *hasVariable* and *hasValue*, which are respectively the variable, and the value to which the variable is bound. It is used to provide a standard evaluation context to the access conditions, e.g., requesting user, resource provider. Consider the following example: L=(<‘‘?resource’’, ‘‘<http://MyExample.net#doc>’’>,<‘‘?user’’,‘‘<http://MyExample.net#sery>’’ >). This list can be used to generate an additional Binding Clause for the access conditions of the form: BINDINGS ?resource ?user {(<http://MyExample.net#doc>, <http://MyExample.net#sery>)}.

Definition 3. *An Access Condition Set (ACS) is a set of Access Conditions.*

The *AccessConditionSet* class has a property *hasAccessCondition* which identifies which Access Conditions form the ACS. Two subclasses of *AccessConditionSet* are introduced: conjunctive, and disjunctive ACS.

Definition 4. *A Conjunctive Access Condition Set (CACS) is a logical conjunction of Access Conditions of the form* $CACS = AC_1 \wedge AC_2 \wedge \ldots \wedge AC_n$. *A CACS is verified if and only if every access conditions it contains is verified.*

Definition 5. *A Disjunctive Access Condition Set (DACS) is a logical disjunction of Access Conditions of the form* $DACS = AC_1 \vee AC_2 \vee \ldots \vee AC_n$. *A DACS is verified if and only if at least one of the access conditions it contains is verified.*

Definition 6. *An Access Tagging Rule (ATR) is a triple* $R = \langle ACS, TagSet, Bindings\rangle$ *where ACS is an Access Condition Set, TagSet is a set of tags* $\{tag_1, tag_2, \ldots, tag_m\}$*, and Bindings is an Access Evaluation Context. An ATR is verified for a named graph tagged with one or more tags from TagSet if and only if the ACS is verified for that named graph.*

An ATR declares that the access conditions in the ACS apply to any named graph tagged with one or more tags from *TagSet*. Notice that the ACS may be reduced to a single access condition. In this case, the ATR is said to be verified if and only if the single access condition is verified. Note that *TagSet* may be empty, in which case the ATR applies to any named graph. The class *AccessTaggingRule* has four properties: *hasAccessConditionSet*, associating an ACS to the ATR, *hasTag*, providing a set of tags to the ATR, *hasAccessEvaluationContext*, associating to the ATR the AEC, i.e., the bindings applied to the rule, and *hasAccessPrivilege*. The *hasAccessPrivilege* property defines the access privilege the user is granted to: *Read*, *Create*, *Update*, *Delete*. We expand the acl:Write class, which is used for every kind of modification on the content, and we allow fine-grained access control privileges. The class *AccessTag*, used to define the set of tags, is a sub-class of scot:Tag.

3.2 The Access Control Policies

We show now which kind of access control policies are enabled by the proposed access control model. Consider the policy defined below: the data provider defines an access policy such that only his named graphs tagged with tag "family" are constrained by the access condition which grants the access to those users which have a *hasParent* relationship with the provider, i.e., the parents of the provider. The Access Condition Set is composed only by one access condition, thus this is the only one which needs to be evaluated. The access privilege is `Update`. Thus, given a `MODIFY` query of the user, if he is granted with the access, then he is allowed to `Update` the requested named graphs. Concerning the contextual information, the Access Tagging Rule grants the access to the user if the date of the access is after December 31^{th} at 23:59. If the user is not granted with the access then the label the system returns him together with the failure message is "parents", to explain that the reasons of the failure have to be associated to the fact that the user is not a parent of the provider; the system does not return the entire policy to the user.

```
<http://MyExample.net/expolicies>
 a s4ac:AccessTaggingRule;
  s4ac:hasAccessConditionSet [
   s4ac:hasAccessCondition [
     s4ac:hasValidity [
       time:hasBeginning [
         time:inXSDDateTime 2011-12-31T23:59:00
       ];
     ];
     s4ac:hasCategoryLabel skos:PrefLabel ''parents''@en;
     s4ac:hasQueryAsk [
       ASK { ?resource dcterms:creator ?provider .
       ?provider rel:hasParent ?user }
     ];
   ];
  ];
  s4ac:hasAccessPrivilege s4ac:Update;
  s4ac:hasTag scot:Tag ''family''@en.
```

The table below presents some examples of the `ASK` queries which may be associated with the access conditions. *Cond1* grants the access to those users who have a relationship of kind "colleagues" with the provider. *Cond2* grants the access to the friends of the provider, and *Cond3* extends this access condition also to the friends of friends. *Cond4* is more complicated[15]. It grants the access to those users that are marked with a specified tag. For specifying the tag, we use the NiceTag ontology. Also negative access conditions are allowed, where we specify which specific user cannot access the data. This is expressed, as shown in *Cond5*, by means of the `FILTER` clause, and the access is granted to every user except *sery*. *Cond6* expresses an access condition where the user can access the data only if he is a minimum lucky, e.g., one chance out of two. Finally, *Cond7* grants the access to those users who are members of at least one group the provider belongs to.

An example of conjunctive ACS is as follows: $CACS_{friends-but-sery} = Cond_2 \wedge Cond_5$, where the access is granted to the users who are friends of the provider, but the user `<http:MyExample.net#sery>`, even if friend of the provider, cannot

[15] The GRAPH keyword is used to match patterns against named graphs.

cond1	`ASK { ?resource dcterms:creator ?provider . ?provider rel:hasColleague ?user }`
cond2	`ASK { ?resource dcterms:creator ?provider . ?provider rel:hasFriend ?user }`
cond3	`ASK { ?resource dcterms:creator ?provider . ?provider rel:hasFriend{1,2} ?user }`
cond4	`ASK { ?resource dcterms:creator ?provider . ?provider dcterms:creator ?g . GRAPH ?g { ?user nicetag:hasCommunitySign ?tag }}`
cond5	`ASK { FILTER(! (?user= <http://MyExample.net#sery>))}`
cond6	`ASK { FILTER(random()>0.5) }`
cond7	`ASK { ?resource dcterms:creator ?provider . ?provider sioc:member_of ?g . ?user sioc:member_of ?g }`

access the data. An example of disjunctive ACS is $DACS_{colleagues-or-friends} = Cond_1 \vee Cond_2$, where it is ensured that the users who are colleagues or friends of the provider are allowed to access the data.

The ATR detailed above can be constrained to a wider set of tags such as $ATR_{parents} = \langle Cond, \{''parent'', ''parents'', ''family'', ''relatives''\}, \emptyset\rangle$ where no AEC is provided. Further examples of ATRs are: (i) $ATR_{friends} = \langle Cond_2, \{''friends'', ''amici'', ''ami''\}, \emptyset\rangle$ where the access condition constrains the access to friends, and three tags are provided without an AEC; (ii) $ATR_{group} = \langle Cond_7, \{''common'', ''group'', ''close''\}, \emptyset\rangle$ is the same for the belonging to the group of the provider; (iii) $ATR_{hiking} = \langle Cond_4, \emptyset, \langle''?tag'', ''hiking''\rangle\rangle$ where the user can access the data if he is tagged with tag "hiking" in the graph created by the provider; (iv) $ATR_{fun} = \langle DACS_{colleagues-or-friends}, \{''fun'', ''funny'', '':-)''\}, \emptyset\rangle$ where the user can access the data if the disjunctive ACS above is satisfied on the named graphs tagged with these three tags.

The prototype under development relies on the SPARQL query engine KGRAM/CORESE[16]. Briefly, the system uses the Binding SPARQL 1.1 to substitute the variable *?resource* with the URI of the named graphs to be accessed. The query is executed to obtain all the ATRs associated with the named graphs, and the data provider. CORESE returns these ATRs which contain the ACS. The `ASK` queries inside the single AC are executed on CORESE, and the returned booleans are conjunctively or disjunctively evaluated to grant or deny the access.

4 Related Work

Sacco and Passant [9] present a Privacy Preference Ontology (PPO), built on top of WAC, in order to express fine-grained access control policies to an RDF file. They also specify the access queries with a SPARQL `ASK`, but their vocabulary does not consider the temporal validity of the privacy preferences, and the number of accesses allowed for each named graph. They rely entirely on the

[16] `http://www-sop.inria.fr/edelweiss/software/corese/`

WAC vocabulary without distinguishing different kinds of `Write` actions. Their model does not allow to specify set of tags to limit the application of the policies to the named graphs marked with those tags, and to specify conjunctive and disjunctive sets of privacy preferences. Muhleisen et al. [8] present a policy-enabled server for Linked Data called PeLDS, where the access policies are expressed using a descriptive language called PsSF, based on SWRL[17]. They distinguish only `Read` and `Update` actions, and they do not consider contextual information. Moreover, the system is based on an ontology of the actions that can be performed on the datasets, i.e., *Action*, *Rule*, *TriplePattern*, no further description is provided in [8].

Giunchiglia et al. [5] propose a Relation Based Access Control model (*RelBAC*), providing a formal model of permissions based on description logics. They require to specify who can access the data, while in our model and in [9] the provider can rely on specifying the attributes the user must satisfy. The Access Management Ontology (AMO) [3] defines a role-based access control model. Such a kind of role-based access control model applied to the world of Linked Data does not provide enough flexibility since it again needs to specify who can access the data. Abel et al. [1] present a model of context-dependent access control at triple level, where also contextual predicates are allowed, e.g., related to time, location, credentials. The policies are not expressed using Web languages, but they introduce an high level syntax then mapped to existing policy languages. They enforce access control as a layer on top of RDF stores. After the evaluation of the contextual information, the queries are expanded, and then sent to the database. Hollenbach and Presbrey [7] present a system where the users can define access controls on RDF documents, and these access controls are expressed using the WAC. Our model extends WAC for allowing the construction of more fine-grained access control policies.

5 Conclusions

In this paper, we introduce a fine-grained model of access control for Linked Data. We rely only on Semantic Web languages, namely SPARQL 1.1 queries, Update language, and Binding Clause. We present the S4AC vocabulary which allows to define various kinds of fine-grained access policies on named graphs. These policies involve both social aspects of the user who wants to access the data, e.g., social relationship with the provider, being member of a group, being tagged with a specific tag, and contextual information, e.g., the day in which the request is performed is in a particular time interval, the user is allowed to access the named graph for five times at most. Policies are evaluated together with a set of tags, which restrain the policies on data tagged in such a way, and an evaluation context which binds the variables of the query to specific values. Moreover, we introduce the four access privileges as defined by the C.R.U.D., and we map them with the SPARQL 1.1 query to identify the policies regarding this privilege which are defined on the requested named graph.

[17] `http://www.w3.org/Submission/SWRL/`

There are different research lines for future work. First, a prototype of the Access Control Manager is under definition together with a user-friendly interface allowing also non-expert users to define their own access terms. Our prototype for the DataLift platform will show a real world application of the proposed model with the aim to test its effectiveness. Second, we plan to introduce delegation in the model, in order to allow the provider to delegate some authority. An open issue remains whether this kind of delegation involves also the authority to modify the access policies defined by the provider. Third, we plan to introduce the licenses, e.g., Creative Commons[18] and Waivers[19], as a further description of the datasets. These licenses then have to be returned together with the requested data, even if the user does not ask explicitly for this information. This is needed to allow data providers to open publish their datasets together with their own terms of reuse.

References

1. Abel, F., De Coi, J.L., Henze, N., Koesling, A.W., Krause, D., Olmedilla, D.: Enabling Advanced and Context-dependent Access Control in RDF Stores. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 1–14. Springer, Heidelberg (2007)
2. Bizer, C., Heath, T., Berners-lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems 5, 1–22
3. Buffa, M., Faron-Zucker, C., Kolomoyskaya, A.: Gestion sémantique des droits d'accès au contenu: l'ontologie AMO. In: Yahia, S.B., Petit, J.M. (eds.) EGC. Revue des Nouvelles Technologies de l'Information, vol. RNTI-E-19, pp. 471–482. Cépaduès-Éditions (2010)
4. Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. J. Web Sem. 3(4), 247–267 (2005)
5. Giunchiglia, F., Zhang, R., Crispo, B.: Ontology driven community access control. In: Proceedings of the 1st Workshop on Trust and Privacy on the Social and Semantic Web, SPOT 2009 (2009)
6. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space, 1st edn. Synthesis Lectures on the Semantic Web: Theory and Technology, vol. 1(1). Morgan & Claypool (2011)
7. Hollenbach, J., Presbrey, J., Berners-Lee, T.: Using RDF Metadata To Enable Access Control on the Social Semantic Web. In: Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge, CK 2009 (2009)
8. Muhleisen, H., Kost, M., Freytag, J.C.: SWRL-based Access Policies for Linked Data. In: Proceedings of the 2nd Workshop on Trust and Privacy on the Social and Semantic Web, SPOT 2010 (2010)
9. Sacco, O., Passant, A.: A Privacy Preference Ontology (PPO) for Linked Data. In: Proceedings of the 4th Workshop about Linked Data on the Web, LDOW 2011 (2011)

[18] http://creativecommons.org/ns

[19] http://vocab.org/waiver/terms/.html

Efficient RDFS Entailment in External Memory

Wouter J. Haffmans and George H.L. Fletcher

Eindhoven University of Technology, The Netherlands
w.j.haffmans@student.tue.nl, g.h.l.fletcher@tue.nl

Abstract. The entailment of an RDF graph under the RDF Schema standard can easily become too costly to compute and maintain. It is often more desirable to compute on-demand whether a triple exists in the entailment. This is a non-trivial task likely to incur I/O costs, since RDF graphs are often too large to fit in internal memory. As disk I/O is expensive in terms of time, I/O costs should be minimized to achieve better performance. We investigate three physical indexing methods for RDF storage on disk, comparing them using the state of the art RDF Schema entailment algorithm of Muñoz et al. In particular, the I/O behavior during entailment checking over these graph representations is studied. Extensive empirical analysis shows that an enhanced version of the state of the art indexing method, which we propose here, yields in general the best I/O performance.

1 Introduction

Data semantics is often defined in terms of inference and reasoning rules. For Resource Description Framework (RDF) data on the web [18], semantics is commonly given using the RDF Schema (RDFS) standard [19], which defines the semantics of web data in terms of inference rules. In particular, given an RDF graph G (i.e., an RDF database instance), application of the rules of RDFS allows us to infer new facts t from G. In this case, we say that G entails t, or that t is in the entailment of G. The union of G and all facts entailed by G is then a new RDF graph called the closure of G. Computing entailment is a fundamental step in many basic RDF reasoning tasks, e.g., [1].

RDFS is known for its complexities involving inferred data [14]. Indeed, the closure of an RDF graph G under RDFS has size quadratic in G, and hence can easily become too costly to compute completely, or to materialize. It is therefore typically more desirable to compute on-demand whether a fact exists in the entailment of G. This is non-trivial requiring traversal of the data in several steps to find possible matches to inference rules. This is likely to incur I/O costs, as RDF graphs are generally too large to fit in main memory and data has to be loaded from external memory. Disk I/O is slow and therefore expensive in terms of time. Hence, the I/O cost should be minimized to achieve acceptable performance.

A variety of physical representations on disk of RDF graphs have been proposed, e.g., [4,8,10,12,13]. Clearly, the choice of storage and indexing strategy

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 464–473, 2011.
© Springer-Verlag Berlin Heidelberg 2011

has a direct impact on I/O behavior. However, to our knowledge, there has been no study to date of the role of this choice in computing RDFS entailment.

In this paper, we study the impact of physical representation on the I/O cost of computing entailment. In particular, we are interested to compare the performance of the state of the art entailment algorithm of Muñoz et al. [7] on the current practice of RDF storage. As this algorithm was only theoretically analyzed by Muñoz et al., we also present here a first empirical use of this algorithm. As such, our investigation led to several practical observations regarding the algorithm, which we discuss below.

We also introduce a novel extension of the Triple-T RDF index [4], the current state of the art, to support entailment checking. Our main contribution in this paper is to show through extensive empirical evaluation that, relative to current proposals for RDF storage, our light-weight extension supports in general the best I/O behavior during entailment computation.

We proceed as follows in the paper. In the balance of this section, we give an overview of related research. When then discuss the approach to entailment checking used in our study in Section 2, followed in Section 3 by a presentation of three approaches to physically representing RDF graphs. In Section 4 we present the design of our experiment, and then give a summary and analysis of the experimental results in Section 5. We close with a summary and discussion of future work in Section 6.

1.1 Related Work

There is a large literature on the performance of different storage methods of RDF data in a variety of ways, for example [4,8,10,12,13]. Much of this research has gone into indexing RDF data for fast query processing, often benchmarked using "wall time" to measure performance. Because large, real-world data sets generally do not fit in main memory, the results are not always guaranteed to provide insight into I/O behavior. In particular, to our knowledge no study has been performed on the I/O performance of indexing techniques for RDFS entailment checking on massive RDF data sets.

There is also a growing body of research on various methods to efficiently compute RDFS entailment. One way is to use a MapReduce technique; the problem is divided and solved in a distributed manner [9,17,20]. In other studies, Datalog engines are used to reason over the entailment of an RDFS graph [6]. Recent work has also studied efficient RDFS inference in a distributed cluster-based setting [11]. Our investigation differs in its focus on the I/O costs of entailment checking over external memory representations of an RDF graph.

2 RDFS Entailment

RDF was devised by the W3C to provide a way of flexibly modeling and sharing data on the web [18]. As the name suggests, it is targeted at describing *resources* (we will also use *atom* as a synonym throughout this paper). This is done by connecting resources in triples of the form (*subject*, *predicate*, *object*). Each resource can be used in any position of a triple, so relations and resources can be easily interchanged. An RDF graph or database is a finite set of such triples.

RDFS is an extension of RDF to provide semantics to resources [14,19]. Whereas RDF provides a basic vocabulary for resources, RDFS adds inferencing rules on top of the vocabulary. Due to these rules, adding a triple to the graph may result in another set of implicit triples also being present in the graph, as discussed above in Section 1. For example, RDFS includes *transitive* behavior for some relations: if A and B are related with a transitive relation R, and resources B and C are related in the same way, then A and C are also related using R. Such implicit data may be quadratic in the size of the original graph, and materializing all this data may be wasteful if it is infrequently used. On the other hand, not materializing the data means the graph has to be scanned for implicit data whenever the data is queried.

In our study we focus on the ρdf subset of RDFS identified by Muñoz et al. [7], capturing the most essential core of RDFS. ρdf is self-contained: it does not rely on the RDFS vocabulary beyond the subset, nor does the rest of the RDFS vocabulary rely on this subset. The resources used in ρdf (with abbreviations) are as follows: `rdfs:subClassOf` (`sc`), `rdf:type` (`type`), `rdfs:subPropertyOf` (`sp`), `rdfs:domain` (`dom`), and `rdfs:range` (`range`).

Following Muñoz et al. [7], we consider graphs G adhering to Minimal RDFS (mrdfs), namely graphs of ground triples which do not redefine the ρdf vocabulary. The deductive rules in mrdfs are determined by taking the rules using ρdf from the original RDFS semantics [18]. We refer the reader to [7] for full details of these rules. Muñoz et al. provide a linearithmic algorithm, excluding reflexivity, to check if a triple exists in the entailment of an RDFS graph [7]. This algorithm is optimal in the sense that any other entailment algorithm must take proportional time [7]. As discussed in the full version of this paper [5], the Muñoz algorithm originally contains one error. For a graph with only triples $(a, \texttt{type}, b)$ and $(b, \texttt{sc}, c)$, the triple $(a, \texttt{type}, c)$ was not found in the entailment. The algorithm was corrected and also extended with reflexivity. The complete algorithm, with these changes is shown in Listing 1.1.

The implementation[1] of the algorithm used in our study contains some further refinements. The first modification is in step 4 of the algorithm. This step first marks many statements in the graph, before checking if there is a path from a specifically marked vertex to the required property p. The proposed change is to start at p and, using G(`sp`), find the subproperties of p. For each of these, as soon as they are found, the algorithm checks if (a, q, b) exists in the graph (with q being the subproperty). This allows the algorithm to terminate as soon as a valid match is found. Additionally, to verify that the traversal terminates, it may be necessary to keep track of which properties were already processed earlier, such that they are not unnecessarily reprocessed. This is to avoid problems in case G contains cycles.

[1] The full software framework is available at `http://www.simply-life.net/rdfsupdate/`.

1. **IF** $p \in \{\texttt{dom}, \texttt{range}\}$ **THEN** check if $(a, p, b) \in \mathrm{G}(\texttt{sp})$
2a. **IF** $p = \texttt{sp}, a \neq b$, **THEN** check if there is a path from a to b in $\mathrm{G}(\texttt{sp})$
2b. **IF** $p = \texttt{sp}, a = b$, **THEN** Check if $a \in \rho$df, if it is not:
Check if one of $\{(?, a, ?), (a, \texttt{sp}, ?), (?, \texttt{sp}, a), (a, \texttt{dom}, ?), (a, \texttt{range}, ?)\}$ is in G
3a. **IF** $p = \texttt{sc}, a \neq b$, **THEN** check if there is a path from a to b in $\mathrm{G}(\texttt{sc})$
3b. **IF** $p = \texttt{sc}, a = b$, **THEN** check if one of
$\{(a, \texttt{sc}, ?), (?, \texttt{sc}, a), (?, \texttt{dom}, a), (?, \texttt{range}, a), (?, \texttt{type}, a)\}$ is in G
4. **IF** $p \notin \rho$df **THEN** check if $(a, p, b) \in \mathrm{G}_\emptyset$: if it is not:
LET $\mathrm{G}(\texttt{sp})^*$ be the graph $\mathrm{G}(\texttt{sp})$ with the following marks:
For each $(u, v, w) \in \mathrm{G}_\emptyset$ then mark v with (u, w).
IN Check in $\mathrm{G}(\texttt{sp})^*$ if there is a path from a vertex marked with (a, b) which reaches p.
5. **IF** $p = \texttt{type}$ **THEN**
LET $\mathrm{G}(\texttt{sp})'$ be the graph $\mathrm{G}(\texttt{sp})$ with the following marks:
For each triple $(u, \texttt{dom}, v) \in G$ mark the vertex u in $\mathrm{G}(\texttt{sp})$ with $d(v)$.
For each triple $(x, \texttt{type}, w) \in G$ mark the vertex x in $\mathrm{G}(\texttt{sc})$ with $t(w)$.
For each triple $(z, e, y) \in \mathrm{G}_\emptyset$ mark the node e with $s(z)$.
LET L_d be the ordered list of elements $d(v)$ such that there is a path from v to b in $\mathrm{G}(\texttt{sc})$.
LET L_t be the ordered list of elements $t(w)$ such that there is a path from w to b in $\mathrm{G}(\texttt{sc})$.
LET L_s be the ordered list of elements $s(z)$ such that either:
(1) in $\mathrm{G}(\texttt{sp})'$ there is a path from a node marked $s(z)$ to a node marked with an element in L_d, or
(2) there is $(z, \texttt{type}, v) \in G$ for $d(v) \in L_d$.
(3) there is $(z, \texttt{type}, w) \in G$ for $t(w) \in L_t$.
IN Check if $\mathrm{s}(a)$ is in L_s.
6. Repeat step 5 symmetrically for `range`, with corresponding modifications

Listing 1.1. Enhanced Muñoz et al. algorithm for checking entailment of triple (a, b, c)

For efficiency reasons, steps 5 and 6 were merged into one step. This allows for constructing L_d (step 5) and L_r (the counterpart of L_d in step 6) in one run. The implementation does not go over $\mathrm{G}(\texttt{sc})$ completely, but only over the superclasses of b. Constructing L_s explicitly is also omitted. Each property in L_d and L_r is used as base. From each such property p', a look-up is performed for $(a, p', ?)$ if p' is in L_d, or $(?, p', a)$ if p' is in L_r. Both look-ups are performed if a property occurs in both lists. From the base, the superproperties are also traversed using the same domain or range role, using similar look-ups.

Whenever a look-up yields a positive result, a is indeed of the requested type b. The algorithm will then terminate early. These modifications do not affect the complexity of the algorithm [5].

3 Physical Representations of RDFS Graphs

In our study, three state of the art methods of physically storing RDFS data sets are compared. Specifically, we compare the currently most used standard

(B-Tree), the state of the art method (Triple-T) and our extension to this method (Triple-T+d).

B^+-Tree Index. The B^+-tree index actually consists of three sub-indexes. The method is widely used in common, commercial RDF storage systems such as *Virtuoso* [3] as well as academic systems such as *RDF-3X* [8]. Although other proposals for indexing have appeared in the research literature, this storage mechanism is currently the most commonly used in practice.

The three sub-indexes make it possible to find triples with one or more unspecified atoms in the query. Each index contains the whole triple, with the atoms in a specific order. The first index, `SOP`, has triples encoded in the order of subject, object and finally predicate. `PSO` encodes the triples in predicate, subject, object order, and finally `OSP` uses the object, subject, predicate order. These orders are determined by the selectivity of the positions in triples [5].

Triple-T Index. The Triple-T index is designed by Fletcher and Beck [4]. One of the key points of this index is that data locality can make joins more efficient. Specific queries are therefore answered more efficiently.

In the original design, the index is built over the space of all atoms. Each index item points to a payload of three buckets. Similarly to the B-Tree index these are in `SOP`, `PSO` and `OSP` orders. The first atom in each of these buckets is left out of the payload, as this is the role of the indexed atom. The buckets are sorted, to make finding matching items possible in $O(log\ n)$ using binary search.

The implemented version has some modifications to the original proposal, as discussed in [5]. A hash table is used to index each atom. Rather than putting all three buckets in the payloads, each atom is indexed up to three times, once for each bucket. For payloads that grow larger than five disk pages, a separate B-Tree is created containing the bucket's data. Payloads are compressed to prevent overflow pages. To make a single implementation of base Triple-T and Triple-T+d feasible, distance information is added to the materialized triples. This information describes the amount of extra (explicit) triples required to derive the triple. In the base Triple-T version, this is always zero.

Triple-T+d Index. This enhanced Triple-T version ("*Triple-T with distance*" or *Triple-T+d*), which we propose here, adds extra information about the distance between subject and object. The basics are all the same as with the regular Triple-T information. The index mainly adds the option to materialize a small part of the closure when updating the index.

The index recognizes two atoms as being transitive: `sp` and `sc`. This follows from rules 2a and 3a in [7]. Additionally the maximum distance to materialize is configurable. For triples with one of the transitive atoms as predicate, the index maintains some inferred triples up to the configured distance. All triples are stored at most once, with the lowest discovered distance. This makes updates more costly, but trying to find a path between two resources using such transitive predicates performs better, as fewer look-ups have to be done in between.

4 Experimental Design

4.1 Test Data

The complexity of entailment checking depends mostly on the ontology size. To properly test the algorithm of Listing 1.1 to the fullest and compare the different indexing methods, synthetic data has to be generated, preferably large sets with a realistic structure. As shown in earlier research, a wide variety of real-world data sets follow a power-law distribution, including many Semantic Web data sets [2,16]. In the sense of RDFS graphs, this means the `sc` and `sp` sub-graphs exhibit a power-law structure.

To generate test data with this structure, the Boost C++ libraries[2] were used – specifically Boost Graph. This library is able to generate graphs following a Power Law Out Degree algorithm. By running this algorithm twice, the required subproperty and subclass relations are created. For simplicity, cycles were removed. To combine the two sub-graphs each property was assigned a domain and range from the generated classes or, for 12.5% of the assignments, an RDF literal. For each class, 10 instance atoms were defined. Each of these is used at least once in a triple, connecting it to another instance atom using a property. Alternatively a direct `type` triple was created, defining the atom as being of the class's type. Further details can be found in the full version of this paper [5].

Several of these data sets were generated. Each one was created using the n parameter of the algorithm as main varying factor. The value for n indicates the amount of classes or properties (in general: graph nodes) to generate. The n values used are 500,000, 1 million, 2 million and 4 million, resulting in graph sizes of 8,028,454; 15,853,402; 32,109,285 and 63,649,886 triples, respectively. More details and statistics of the data sets can be found in [5].

4.2 Test Queries

After creating the data sets, it became possible to extract queries from these sets. There are various distinct steps in Listing 1.1, of which at any time just one step is executed (taking into consideration that steps 5 and 6 are combined to one step).

To compare the physical representations with different use cases, three classes of queries were constructed. The first class (Class 1) is a simple look-up. It executes step 1 by checking for a triple of the form $(?, \texttt{dom}, ?)$ or $(?, \texttt{range}, ?)$, of which the subject is a known property and the object a known class.

The second class (Class 2) is a single-step traversal using `sp`. First the disconnected `sp`-triples were found. These are triples $(A, \texttt{sp}, B)$ for which A and B are not used in any other `sp`-triples. For a random selection of these, a random triple of the form (P, A, Q) is looked up. Subject P and object Q in such triples are instance atoms. The query then becomes (P, B, Q). This is a direct application of step 3b.

[2] Boost C++ Libraries: http://www.boost.org/

The third class (Class 3) of queries are those following implicit typing. The longest chains of sp-triples are located in the graph, and the domain and range of the last property (the super-property of all others) are located. Of these classes identified by the domain and range, the longest available sc chains are located in a similar way. These chains start with a subclass, going up to a root superclass using sc-triples. Next an instance of the property at the start of the sp-chain is located, and the final query is generated of the form ($ATOM$, type, $CLASS$).

4.3 Implementation and Execution

All three indexes are implemented based on an abstract interface. The algorithm of Listing 1.1 uses this interface directly to perform look-ups. Each step of the algorithm is implemented separately, with several generalizations as discussed in [5]. This includes doing a look-up for the query at the start of the algorithm, regardless of its form. To ensure that all metrics are comparable, each index is implemented using Berkeley DB.[3]

A moderate desktop PC was used to run the experiments. The queries were executed 100 times in a row on the corresponding data set, using each physical index. In between each query, the Berkeley DB cache was flushed by closing the database, removing the cache files and re-opening the database. The cache size of Berkeley DB was set to 1 Mb, the page size to 4 Kb, for all experiments. The amount of cache misses as reported by Berkeley DB are compared as indicator of the I/O costs of each index. Triple-T+d was configured to materialize a single extra step.

The full experimental procedure was as follows. Each data set was loaded and 100 queries of each class were generated using the given data set. Half of these 100 were "positive" queries: the triples are known to exist in the data set. The other 50 are "negative" queries. These triples do not exist in the entailment of the data set. The negative queries were created by replacing the subject or object in an positive query with another (randomly chosen) resource of a similar type. As such the form of the query is the same. These replacements were chosen such that the complexity of answering the query is kept high, for example by using data of sp or sc chains with similar length.

5 Empirical Results and Analysis

We next discuss the results of the experiments.

Database sizes. A significant issue in storage and indexing is the space occupied on disk. Comparing the database sizes of the three indexes showed Triple-T is always the smallest (on average 4.2% smaller than B-Tree), and that Triple-T+d consumes the most space on the smallest two data sets, but is slightly smaller than B-Tree in the largest data sets. On average over all sets, Triple-T+d consumes 4.7% more space than Triple-T, and 0.3% more than B-Tree.

[3] Berkeley DB: http://www.oracle.com/technetwork/database/berkeleydb

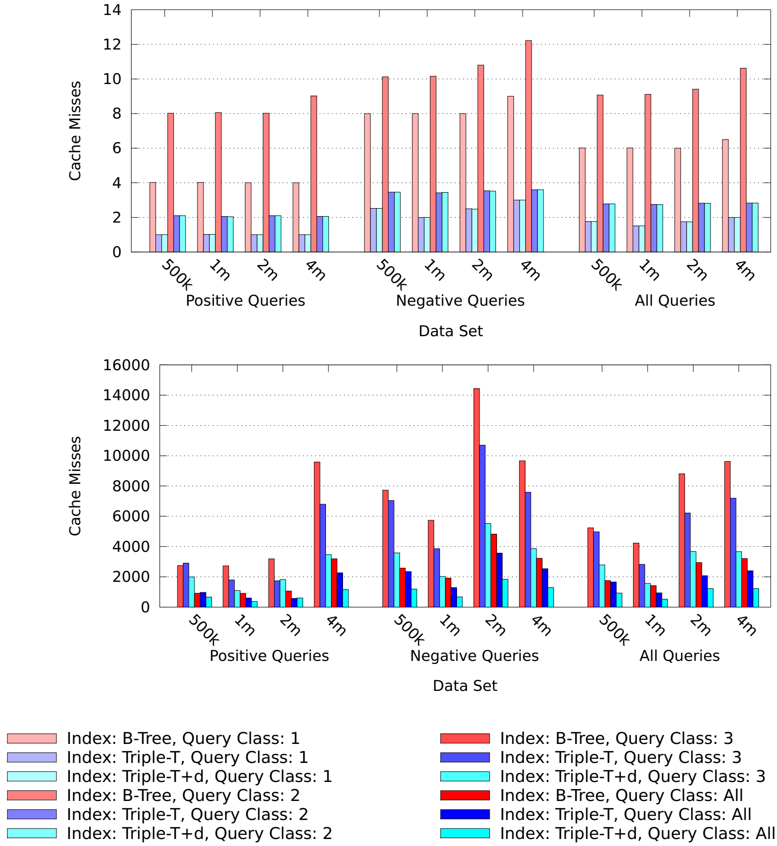


Fig. 1. Average Cache Misses per Query

I/O performance. The I/O costs of entailment checking, measured in terms of average cache misses per query, are presented in Figure 1. As expected, Triple-T and Triple-T with distance perform equally well on the two base query classes. The extra data that is materialized in the enhanced version is not used for these queries. For complex queries, it becomes clear that adding the extra data has its benefits. Whereas the basic Triple-T has about as many cache misses as the B-Tree solution, this amount is almost always lower when a small part of the entailment is materialized.

The only exception are the positive queries in the *2m* set, where base Triple-T performs slightly better than the version with distance. This may be due to characteristics of the queries or the data set. It is possible that several queries could be answered without any traversal, in which case the enhanced version may require more data to be loaded.

Additionally, in the *500k* data set, while the total amount of cache requests for basic Triple-T is lower than the B-Tree set, the amount of cache misses is slightly higher. This can be caused by the fact that for each look-up, Triple-T has to load payload data – which may include data that is not immediately required. This causes the cache to fill more quickly, which makes the working set smaller. As such, it may happen that some data needs to be re-loaded during the execution of a query. Increasing the cache size slightly may therefore reduce this difference, or even make Triple-T perform better than B-Tree. Triple-T with distance does perform better than both other indexes in this case.

In summary, the experiments show that our proposed Triple-T+d structure is both space-efficient as well as I/O efficient for entailment queries.

Summary of other results. The full version of this paper [5] discusses further empirical results. Further data and query sets were generated using the PoweR-Gen benchmark [15]. Also, the total wall-clock time was measured using a cold OS cache. Generally Triple-T+d has shown to perform the best for these results as well. Finally a comparison was made when changing the `OSP` order in the indexes into `OPS`, which was expected to make look-ups with known object and predicate more efficient in some scenarios. However, there was no clear winner in this experiment.

6 Concluding Remarks

Motivated by the growing size of semantic web data sets and the common use of RDFS to define their semantics, in this paper we studied the problem of efficiently computing RDFS entailment in external memory. The state of the art entailment algorithm of Muñoz et al. was first empirically validated and enhanced. Two state of the art physical representations of RDF data and one novel representation proposed here were compared using this enhanced algorithm over massive test data sets which model the structure of real-world data. Extensive empirical analysis showed that our proposed indexing approach was both space and I/O efficient, with respect to the state of the art.

There are many interesting directions for further investigation. We close by listing a few of these. First, further enhancements to Triple-T+d should be investigated, as here we have only considered one natural way to extend the Triple-T structure. Alternative approaches to materializing the transitive sub-graphs may make it possible to find paths between nodes even more quickly and efficiently, further reducing I/O costs. Secondly, the entailment algorithm can be extended with support for blank nodes, as discussed in [7], or with support for richer rules sets. Finally, applying the proposed approach to efficient external memory entailment to support other RDF reasoning tasks, such as handling graph updates under RDFS semantics [1], is a natural next step towards practical management of massive RDF graphs.

Acknowledgments. We thank Paul De Bra and Christian Stahl for their many helpful comments on this investigation.

References

1. Chirkova, R., Fletcher, G.H.L.: Towards well-behaved schema evolution. In: WebDB (2009)
2. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Review 51(4), 661–703 (2009)
3. Erling, O.: Towards web scale RDF. In: SSWS (2008)
4. Fletcher, G.H.L., Beck, P.W.: Scalable indexing of RDF graphs for efficient join processing. In: CIKM, pp. 1513–1516 (2009)
5. Haffmans, W.: A Study of Efficient RDFS Entailment in External Memory. Master's thesis, Eindhoven University of Technology (June 2011), http://www.simply-life.net/thesis.pdf
6. Ianni, G., Krennwallner, T., Martello, A., Polleres, A.: Dynamic Querying of Mass-storage RDF Data with Rule-based Entailment Regimes. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 310–327. Springer, Heidelberg (2009)
7. Muñoz, S., Pérez, J., Gutierrez, C.: Simple and efficient minimal RDFS. J. Web Semantics 7(3), 220–234 (2009)
8. Neumann, T., Weikum, G.: The RDF-3X engine for scalable management of RDF data. VLDB J. 19(1), 91–113 (2010)
9. Oren, E., Kotoulas, S., Anadiotis, G., Siebes, R., ten Teije, A., van Harmelen, F.: Marvin: Distributed reasoning over large-scale semantic web data. J. Web Semantics 7(4), 305–316 (2009)
10. Sakr, S., Al-Naymat, G.: Relational processing of RDF queries: a survey. SIGMOD Record 38(4), 23–28 (2009)
11. Salvadores, M., Correndo, G., Harris, S., Gibbins, N., Shadbolt, N.: The Design and Implementation of Minimal RDFS Backward Reasoning in 4store. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 201. LNCS, vol. 6644, pp. 139–153. Springer, Heidelberg (2011)
12. Schmidt, M., Hornung, T., Küchlin, N., Lausen, G., Pinkel, C.: An Experimental Comparison of RDF Data Management Approaches in a SPARQL Benchmark Scenario. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 82–97. Springer, Heidelberg (2008)
13. Sidirourgos, L., et al.: Column-store support for RDF data management: not all swans are white. PVLDB 1(2), 1553–1563 (2008)
14. ter Horst, H.J.: Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the OWL vocabulary. J. Web Semantics 3(2-3), 79–115 (2005)
15. Theoharis, Y.: On Power Laws and the Semantic Web. Master's thesis, University of Crete (February 2007)
16. Theoharis, Y., Tzitzikas, Y., Kotzinos, D., Christophides, V.: On graph features of semantic web schemas. IEEE Trans. Knowl. Data Eng. 20(5), 692–702 (2008)
17. Urbani, J., Kotoulas, S., Oren, E., van Harmelen, F.: Scalable distributed reasoning using mapReduce. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 634–649. Springer, Heidelberg (2009)
18. W3C: RDF semantics (February 2004), http://www.w3.org/TR/rdf-mt/
19. W3C: RDF vocabulary description language 1.0: RDF schema (February 2004), http://www.w3.org/TR/rdf-schema/
20. Williams, G.T., Weaver, J., Atre, M., Hendler, J.A.: Scalable reduction of large datasets to interesting subsets. J. Web Semantics 8(4), 365–373 (2010)

Linked Data Indexing Methods: A Survey*

Martin Svoboda and Irena Mlýnková

Department of Software Engineering, Charles University in Prague
Malostranske namesti 25, 118 00 Prague 1, Czech Republic
{svoboda,mlynkova}@ksi.mff.cuni.cz

Abstract. Documents on the contemporary Web are based especially on HTML formats and, therefore, it is rather difficult to retrieve hidden structured information from them using automated agents. The concept of Linked Data based primarily on RDF data triples seems to successfully solve this drawback. However, we cannot directly adopt the existing solutions from relational databases or XML technologies, because RDF triples are modelled as graph data and not relational or tree data. Despite the research effort in recent years, several questions in the area of Linked Data indexing and querying remain open, not only since the amount of Linked Data globally available significantly increases each year. This paper attempts to introduce advantages and disadvantages of the state-of-the-art solutions and discuss several issues related to our ongoing research effort – the proposal of an efficient querying framework over Linked Data. In particular, our goal is to focus on large amounts of distributed and highly dynamic data.

Keywords: Linked Data, RDF, indexing, querying, SPARQL.

1 Introduction

The majority of documents on the contemporary Web are based primarily on HTML formats. Although these documents often contain hidden structured and interlinked information, it is quite difficult for automated agents to retrieve such information. Therefore, an idea of Linked Data appeared in order to extend the Web of Documents towards the Web of Data [4].

Linked Data do not represent any particular standard; we only talk about a set of recommended principles and techniques, which lead to the publication of data in a way more suitable for their automated processing. First, each real-world entity should be described by a unique URL identifier. These identifiers can be dereferenced by HTTP to obtain information about the given entities. And, finally, these entity representations should be interlinked together to form a global open data cloud – the Web of Data.

Even though there are several particular ways, the most promising is probably the RDF (Resource Description Framework) [15] standard, where data are

* This work was supported by the Charles University Grant Agency grant 4105/2011, the Czech Science Foundation grant P202/10/0573 and the grant SVV-2011-263312.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 474–483, 2011.
© Springer-Verlag Berlin Heidelberg 2011

modelled as triples conforming to the concept of *subject-predicate-object*. An alternative way to view these triples are graphs, where vertices correspond to *subjects* and *objects*, edges represent the triples themselves and are labelled by *predicates*. At the implementation level we can publish RDF triples in a form of RDF/XML [3] syntax and along the data we can also publish RDFS [6] schemata or OWL [16] ontologies restraining the allowed content of such RDF data.

In recent years, a significant effort was made not only in a theoretical research, but also in the amount of Linked Data globally available. However, since RDF triples are modelled as graphs, we cannot directly adopt existing solutions from relational databases and XML [5] technologies. Thus, the area of Linked Data includes many open problems, starting from application architecture questions and ending, e.g., with user interaction paradigm.

Objectives. The purpose of this paper is to provide a survey of the area of Linked Data indexing techniques. It is clear that indexing is tightly connected with storing and querying, however, we will only focus on state-of-the-art solutions in the area of indexing. We will mutually compare the existing approaches, discuss general issues and important aspects of a proposal of index structures supporting effective and efficient querying, but we will not provide any performance comparisons. Thus, this paper should give the reader the basic insight into the area of Linked Data indexing, its various ideas and principles, but due to the lack of space, we are not able to go to detail.

Outline. In section 2 we discuss issues and dimensions that need to be considered in the comparison of Linked Data indexing methods. Section 3 gives the thorough description of the existing approaches in this area, while in Section 4 we provide an overall summary of these approaches and discuss found observations and open questions. Finally, Section 5 concludes this paper.

2 Analysis

Whereas the logical model of RDF data is based on graphs, standard databases work with relational tables and semi-structured XML documents can be viewed as trees. Despite these differences, we can still find inspiration in these two well established areas. In this section, we outline a number of issues that need to be considered when proposing new indexing methods. We need to consider at least architecture, storage and querying aspects. Next, we introduce a set of dimensions using which we can compare and classify the existing approaches.

2.1 General Issues

Architecture. The fundamental question of each querying system is the mutual relationship between physical storages, index structures and querying capabilities. It is clear that querying without data has no sense, and vice versa. In other words, all presented issues can only hardly be considered separately. First, we need to discuss what querying constructs we need in a particular application, i.e. what expressive power we require or even which querying language we want to choose. Then, the storage itself and index structures should be proposed in the way capable to effectively and efficiently provide results of queries.

Since the concept of Linked Data emerged to support the idea of the Web of Data, we cannot ignore its main feature based on data distribution. Having no central points in the infrastructure would be worth, but it seems that without at least some local knowledge, the querying over the Web would not be sufficiently fast. If we want to evaluate queries only via online accessing remote data sources, we are assured that we obtain accurate and up-to-date results, but we cannot neglect the amount of data needed to be transferred and the required time.

On the other hand, local approaches may enable more efficient query processing, but we need to deal with large amounts of data and their ageing. It seems that a sort of federated and integrated querying framework [21] could be a good way to overcome the mentioned issues. Nevertheless, we need to consider several reasons that may prevent maintaining local data copies. Besides technical ones, there can also be legal reasons such as copyrights.

Storage. In standard relational databases, data are stored in tables and index structures have only an auxiliary role to support efficient query evaluation. The important thing is that no data are stored in indices themselves, since they only contain duplicated data fragments or derived statistics. In the area of RDF storing, we can even find indexing approaches [2] which are completely isolated from the physical database layer, i.e. a query can be evaluated only using the index structure itself.

Although native approaches for querying RDF data could generally represent more efficient solutions, the mentioned relational databases [1] benefit from decades of experience and research results. We can come across three basic ways how to store RDF data in relational tables. The first possible solution is based on one big table with three columns for subjects, predicates and objects. The second approach is based on a set of tables, where values in the first column always identify subjects and values in remaining columns correspond to objects of predicates determined by these columns (we group several predicate and object pairs for a given subject). Finally, the third approach has a separate table for each predicate, always with 2 columns for subject and object pairs.

Querying Language. According to [7] we can divide RDF querying languages into three basic layers: syntactic, structural and semantic. Probably the most widespread language SPARQL [20] represents the structural querying. Having a query formulated as a graph pattern with fixed values and/or variables, the query processor attempts to find all matches of this graph pattern against the entire data graph stored in the database.

The important position has also the concept of fulltext querying based on keywords. Although we may require such functionality even in local querying systems, these techniques play the main role in Web search engines. The important feature of this model is that we are usually not interested in which particular component (subject, predicate, object) the specified keyword should exactly be located, whereas in graph querying models the value equality testing actually needs to consciously distinguish between particular triple particles.

After a submitted query is parsed and transformed into an internal representation, we need to find the most suitable query evaluation plan. In this sense a special position in query optimization techniques has the join ordering. It is quite interesting that we can use similar ideas to the nested loop algorithm from relational databases. However, there are other aspects that need to be considered. The problem is not only that we usually have to rely on heuristics, since we are not able to consider all possible plans, but we also use imprecise statistics.

2.2 Classification of Approaches

Existing approaches can be compared in many different ways. First of all, we can talk about the architecture, which is primarily derived from the scope of processed data. We distinguish between *local*, *distributed* and *global* approaches.

If the index structures themselves allow efficient insertions, updates and deletions, we talk about *dynamic* structures, otherwise about *static* ones. These structures may serve for indexing *data*, or only *statistics* about them. Anyway, each system focuses on different data units: *triples* conform to standard RDF triples, *quads* are extended with the context, whereas in case of *sources* we work with semantic documents or other files and services.

Approaches may also be compared by querying languages, their expressive power or allowed constructs. Although there are *syntactic* and *semantic* layers, we only focus on the *structural* one. In this case, queries can be based on *fulltext* or *graph* patterns. Anyway, the querying model is closely connected with the model of index structures. Thus, we can index *keywords*, *triples*, *quads*, *trees*, *paths* or other *areas*. Indexing approaches also differ in access patterns: *universal* treat subject, predicate and object components equally, whereas *dedicated* not.

3 Approaches

Not all possible combinations derived from introduced dimensions make sense. Nevertheless, we are able to derive three main categories of existing approaches: *local* querying systems, *distributed* source selection techniques and *global* searching engines. The description of existing approaches is the subject of this section. For simplification, we will use abbreviations S, P, O and C for *subject*, *predicate*, *object* and *context* quad particles respectively.

3.1 Local Approaches

Quad Index. Index structures proposed by Harth and Decker [11] enable querying of local data quads with context. These structures involve Lexicon (an inverted list for keywords and two-way translation maps for term identifiers based on B^+-trees) and Quad indices (B^+-trees for $SPOC$, POC, OCS, CSP, CP, OS orderings) allowing to query in all 16 access patterns. Despite data quads themselves, these indices also contain statistics about data, e.g. quad $(s, p, 0, 0)$ in $SPOC$ index represents the number of all quads with given s and p values.

Vertical Partitioning. Abadi et al. [1] proposed a model of storing RDF triples in relational databases in a way of a separate table for each predicate (with one column for subjects, second for objects). This implies that we are able to store pre-computed paths. Similarly to the previous approach, the authors use the translation of strings into identifiers.

RDF-3X Engine. The core of the stream processor RDF-X proposed by Neumann and Weikum [17] is based on six B^+-tree indices for all SPO, SOP, OSP, OPS, PSO and POS access patterns. Additionally, they also use indices with statistics (S, P, O, SP, PS, PO, OP, SO and OS projections) and selectivity histograms and statistics for pre-computed path or star patterns.

Sextuple Index. The idea of HexaStore approach by Weiss et al. [27] is based on similar SPO, SOP, OSP, OPS, PSO and POS index structures, however, these are implemented as ordered nested lists. All these lists contain only identifiers instead of strings, again.

Matrix Index. BitMat is an approach proposed by Atre et al. [2]. The index model is based on a matrix with three dimensions for S, P and O values (terms are translated to identifiers, which are used as matrix indices). Each cell contains a bit value equal to 1 if and only if the given triple is stored in the database, otherwise value 0. The index is organized as an ordinary file with all SO, OS, PO and PS slices stored using a bit run compression over individual slice rows.

Path Index. An index structure for path queries proposed by Liu and Hu [14] is inspired by traditional information retrieval methods for processing texts. Its main idea is based on suffix arrays, using which we are able to efficiently index paths as sequences of fixed S, P and O values.

GRIN Index. Udrea et al. [26] introduced a model based on splitting data graphs into subgraph areas that are described by conditions limiting their content. The idea is derived from a metric defined on URIs and literals (e.g. a minimal number of edges in a data graph between a given pair of values). The index structure itself is a balanced binary tree, where internal nodes represent mentioned areas and leaf nodes store data triples conforming to these areas.

Structure Index. The last presented local approach is a parameterised index introduced by Tran and Ladwig [25]. Their model is based on bisimilarity relations, putting in a relation such two vertices of the data graph that share the same outgoing and ingoing edges (reflecting only predicates). Vertices from the same equivalence class have the same characteristics and, therefore, prompted queries can first be evaluated over these classes to prune required data.

3.2 Distributed Approaches

The purpose of the following approaches is to work with distributed sources and provide transparent querying over their data.

Repository Index. An index by Stuckenschmidt et al. [22] is inspired by object databases. It captures statistics about paths and enables querying over them using tree pattern queries with fixed values of predicates. The index structure itself is hierarchical (for each path it also contains all its subpaths).

Federated Querying. Quilitz and Leser [21] proposed a system for transparent integrated querying over distributed and autonomous sources. The core of this approach is a language for description of distributed sources, in particular, data triples they contain, together with other source characteristics.

Data Summaries. The purpose of a data summary index by Harth et al. [12] is to enable the source selection over distributed data sources. Data triples are modelled as points in a 3-dimensional space (S, P, and O coordinates are derived by hash functions). The index structure is a QTree based on standard R-Trees. Internal nodes act as minimal bounding boxes for nested nodes, leaf nodes contain statistics about data sources, not data triples themselves.

3.3 Global Approaches

Finally, we briefly outline three global searching approaches. All of them are primarily inspired by traditional information retrieval methods.

Swoogle. The first system was proposed by Ding et al. [9]. Its objective is to serve as a searching engine over semantic documents, both data and ontologies.

SWSE. The purpose of SWSE by Harth et al. [10] is to provide a system for global searching over quads (RDF triples with their context). The querying focuses not only on keyword matching, but it also supports concept filtering.

Sindice. Oren et al. [18] introduced a global engine for searching semantic documents on the Web, allowing to query via keywords, inverse functional properties and resource URIs.

4 Summary

Summarising approach descriptions in the previous section, we can outline several observations and aspects that we consider as important to be discussed during the proposal of new techniques for indexing Linked Data. We have also identified several areas we consider as not yet sufficiently investigated.

4.1 Approaches Comparison

It seems that the existing approaches represent efficient proposals for querying RDF data. However, the majority of existing approaches have several limitations or assumptions which cause their difficult usage in the general case. Moreover, their efficiency often highly depends on allowed querying constructs, but it is not clear, what characteristics real-world data and queries really have.

In Table 1 we provide a brief and simplified overview of all existing approaches we have listed in this paper. Besides different intended purposes of these proposals, we can especially put the attention to the comparison of data, query and index models behind them.

Despite the ideas of particular solutions are sometimes very different, we can find at least the following common aspects.

Table 1. Comparison of existing approaches

Approach name	Scope type	Data items	Query model	Index model	Basic approach description
Local Querying Methods					
Harth 2005 Quad index [11]	Local	Quads	Graphs YARS QL	Quads Text	Lexicon using inverted lists Quad indices using B^+-trees
Abadi 2007 Partitioning [1]	Local	Triples	Graphs SQL	Paths	Table for each predicate Standard RDBMS indices
Neumann 2008 RDF-3X [17]	Local	Triples	Graphs SPARQL	Triples	Triple, double, single B^+-trees Histograms and path statistics
Weiss 2008 Hexastore [27]	Local	Triples	Graphs SPARQL	Triples	Nested ordered lists Dictionary encoding of URIs
Atre 2010 BitMat index [2]	Local	Triples	Graphs SPARQL	Triples	Slices of 3D matrix Bit run compression of rows
Liu 2005 Path index [14]	Local	Triples	Paths SPARQL	Paths	Paths as sequences of terms Stored using suffix arrays
Udrea 2007 GRIN index [26]	Local	Triples	Graphs SPARQL	Circles	Unlimited paths querying Binary tree over graph circles
Tran 2010 Struct. index [25]	Local	Triples	Graphs SPARQL	Graphs	Contraction to extensions Sets of in/outgoing predicates
Distributed Querying Methods					
Stucken. 2004 Repositories [22]	Dist.	Sources	Trees SeRQL	Paths	Paths as predicate sequences Hierarchical paths index
Quilitz 2008 DARQ [21]	Dist.	Sources	Graphs DARQ	Services	Service descriptions index Capabilities and selectivity
Harth 2010 Summaries [12]	Dist.	Sources	Graphs SPARQL	Boxes	Triples hashing to 3D points Q-tree index with buckets
Global Searching Methods					
Ding 2004 Swoogle [9]	Global	Files	Fulltext	Text	Semantic web documents Keywords and n-grams
Harth 2007 SWSE [10]	Global	Quads	Fulltext	Text	Global quads querying Keywords, IFP and concepts
Oren 2008 Sindice [18]	Global	Files	Fulltext	Text	Invertes lists to sources Keywords, URI and IFP

Compression. The shared idea of the majority of indexing methods is the way of storing string values of URIs and literals, since there is a high probability that strings may have multiple occurrences in the database. Therefore, it seems very effective to store these strings only once in a special storage, assign them unique integer identifiers, and use them in RDF triples instead of the original terms. As a consequence, frequently executed value equality tests during the query evaluation may then be executed much faster.

Data Pruning. Efficient systems also support the query evaluation via a set of optimizations. We can propagate data filtering selections as close as possible to their fetching, or we can perform data pruning before the phase of joining. In case of distributed approaches, we focus on the problem of source selection, i.e. to access data only of those remote sources that are relevant to the query. Generally, we want to avoid processing of irrelevant data whenever possible.

4.2 Open Problems

According to the discussed issues and the comparison of the existing approaches, we can formulate the following open problems in the area of indexing.

Architecture. Finding an appropriate compromise between processing local or distributed data forms one of the most important questions. Maintaining local copies of data may benefit from convenient conditions for more efficient query evaluation; however, we are not always able or allowed to gather the data under our control. The second approach is based on accessing distributed data on-the-fly using link traversal. This method suffers from transfer requirements, although it ensures work with up-to-date data.

Scalability. Even though existing approaches work with large sets of data, experiments performed using various sets of data, queries and prototype implementations of discussed solutions imply that we are still not able to sufficiently flatten performance of such approaches and the explosion of the Web of Data size. While we could find about 10 globally important data sources with 920 million triples and 150 thousand links in 2007[1], these numbers increased to about 200 important sources, 25 billion triples with 395 million links in 2010[2].

Dynamicity. Although experiments [8] consider the Web of Documents, it seems that data on the Web of Data tends to aging too. Moreover, we especially need not only to handle simple data modifications, but also deal with broken links and attempt to anticipate or correct them. Achieving consistently connected data [19] is necessary, when we maintain local data copies or summaries. Unfortunately, the problem is that index structures are often static and do not allow any further modifications like inserts, updates or deletes.

Quality. The increasing number of globally available data on the Web also causes issues of data quality and trust. Especially in the context of global search engines we need to propose accurate metrics or other techniques for determining relevance of particular query answers. For this purpose we can utilize data provenance or even knowledge and relationships from social networks [13].

1 http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData/
2 http://www.ckan.net/

5 Conclusion

Recent years show that Linked Data became a suitable way of building the Web of Data that can be automatically processed without any special effort, such as without complicated information retrieval from standard HTML pages on the contemporary Web of Documents.

Since Linked Data in a form of the RDF standard represent graph data, we cannot directly adopt existing research results from areas of relational databases and XML technologies. And since the amount of Linked Data globally available still grows, several questions need to be solved to offer efficient systems.

The purpose of this paper was to provide an overview of existing Linked Data indexing approaches. We discussed issues related to the proposal of new indexing methods, introduced a set of dimensions for comparing existing solutions, and also identified aspects that can still be considered as open problems.

This survey is connected with our ongoing research effort [23] that should result in a proposal of an entire framework for efficient and effective querying of Linked Data. In particular, we want to deal especially with the following aspects: data *distribution*, *scalability*, *dynamicity* and *quality*. Although existing approaches show promising ways of solving our problem, the combination of all named assumptions remains unsolved. For this purpose we not only want to propose novel techniques together with the prototype implementation, but we also want to harness characteristics of real-world data detected using Analyzer [24], our framework for robust analyses of documents on the Web.

References

1. Abadi, D.J., Marcus, A., Madden, S.R., Hollenbach, K.: Scalable Semantic Web Data Management Using Vertical Partitioning. In: Proc. of the 33rd Int. Conf. on Very Large Data Bases, VLDB 2007, pp. 411–422. VLDB Endowment (2007)
2. Atre, M., Chaoji, V., Zaki, M.J., Hendler, J.A.: Matrix "Bit" loaded: A Scalable Lightweight Join Query Processor for RDF Data. In: Proc. of the 19th Int. Conf. on World Wide Web, WWW 2010, pp. 41–50. ACM, New York (2010)
3. Beckett, D.: RDF/XML Syntax Specification, Revised (2004), `http://www.w3.org/TR/rdf-syntax-grammar/`
4. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
5. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F., Cowan, J.: Extensible Markup Language (XML) 1.1, 2nd edn. (2006), `http://www.w3.org/TR/xml11/`
6. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema (2004), `http://www.w3.org/TR/rdf-schema/`
7. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
8. Cho, J., Garcia-Molina, H.: The Evolution of the Web and Implications for an Incremental Crawler. In: Proc. of the 26th Int. Conf. on Very Large Data Bases, VLDB 2000, pp. 200–209. Morgan Kaufmann Publishers Inc., USA (2000)

9. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: A Search and Metadata Engine for the Semantic Web. In: Proceedings of the 13th ACM Int. Conference on Information and Knowledge Management, CIKM 2004, pp. 652–659. ACM, New York (2004)
10. Harth, A., Hogan, A., Delbru, R., Umbrich, J., O'Riain, S., Decker, S.: SWSE: Answers Before Links. In: Proc. of the Semantic Web Challenge 2007 co-located with ISWC 2007 + ASWC 2007, vol. 295, pp. 136–144. CEUR-WS.org (2007)
11. Harth, A., Decker, S.: Optimized Index Structures for Querying RDF from the Web. In: Third Latin American Web Congress, LA-WEB 2005, IEEE (2005)
12. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.U., Umbrich, J.: Data Summaries for On-demand Queries over Linked Data. In: Proc. of the 19th Int. Conf. on World Wide Web, WWW 2010, pp. 411–420. ACM, NY (2010)
13. Knap, T., Mlynkova, I.: Quality Assessment Social Networks: A Novel Approach for Assessing the Quality of Information on the Web. In: QDB, pp. 1–10 (2010)
14. Liu, B., Hu, B.: Path Queries Based RDF Index. In: Proceedings of the First International Conference on Semantics, Knowledge and Grid, pp. 91–93. IEEE Computer Society, Los Alamitos (2005)
15. Manola, F., Miller, E.: RDF Primer (2004), http://www.w3.org/TR/rdf-primer/
16. McGuinness, D.L., Harmelen, F.v.: OWL Web Ontology Language: Overview (2004), http://www.w3.org/TR/owl-features/
17. Neumann, T., Weikum, G.: RDF-3X: A RISC-style Engine for RDF. Proc. VLDB Endow. 1, 647–659 (2008)
18. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: Sindice.com: A Document-oriented Lookup Index for Open Linked Data. International Journal of Metadata, Semantics and Ontologies 3(1), 37–52 (2008)
19. Popitsch, N.P., Haslhofer, B.: DSNotify: Handling Broken Links in the Web of Data. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 761–770. ACM, New York (2010)
20. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF (2008), http://www.w3.org/TR/rdf-sparql-query/
21. Quilitz, B., Leser, U.: Querying Distributed RDF Data Sources with SPARQL. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 524–538. Springer, Heidelberg (2008)
22. Stuckenschmidt, H., Vdovjak, R., Houben, G.-J., Broekstra, J.: Index Structures and Algorithms for Querying Distributed RDF Repositories. In: Proc. of the 13th Int. Conf. on World Wide Web, WWW 2004, pp. 631–639. ACM, NY (2004)
23. Svoboda, M., Mlynkova, I.: Efficient Querying of Distributed Linked Data. In: Proceedings of the 2011 Joint EDBT/ICDT Ph.D. Workshop, PhD 2011, pp. 45–50. ACM, New York (2011)
24. Svoboda, M., Stárka, J., Sochna, J., Schejbal, J., Mlýnková, I.: Analyzer: A Framework for File Analysis. In: Yoshikawa, M., Meng, X., Yumoto, T., Ma, Q., Sun, L., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 6193, pp. 227–238. Springer, Heidelberg (2010)
25. Tran, T., Ladwig, G.: Structure Index for RDF Data. In: Workshop on Semantic Data Management (SemData@VLDB) 2010 (2010)
26. Udrea, O., Pugliese, A., Subrahmanian, V.S.: GRIN: A Graph Based RDF Index. In: Proceedings of the 22nd National Conference on Artificial Intelligence, vol. 2, pp. 1465–1470. AAAI Press (2007)
27. Weiss, C., Karras, P., Bernstein, A.: Hexastore: Sextuple Indexing for Semantic Web Data Management. Proc. VLDB Endow. 1, 1008–1019 (2008)

Noise Reduction in Essay Datasets for Automated Essay Grading

Anhar Fazal, Tharam Dillon, and Elizabeth Chang

Digital Ecosystems and Business Intelligence Institute
Curtin University of Technology
Perth, Australia
anhar.fazal@postgrad.curtin.edu.au,{tharam.dillon,
elizabeth.chang}@cbs.curtin.edu.au

Abstract. Marking of a huge number of essays is a very burdensome and tedious task for the teacher and/or trainer. Studies have shown that their efficiency decreases significantly when continuously marking essays over a given period of time. An Automated Essay Grading (AEG) system would be most desirable in such a scenario to reduce the workload of the teacher and/or trainer and to increase the efficiency of the marking process. Almost all the existing AEG systems assume that the relationship between the features of the essay and the essay grade is linear, which may not necessarily be the case. In cases where the relationship between the feature vector and the essay grade is non-linear, none of the existing methods provides a mechanism to capture that and determine an accurate essay grade. This paper proposes a new AEG system, the OzEgrader, that aims to capture both the linear and non-linear relationships between the essay features and its grade, and explains the methodology for noise reduction in the essay dataset.

Keywords: Automated Essay Grading, Noise reduction, Natural Language Processing, Statistical methods and Hybrid methods.

1 Introduction

Manual grading of essays is time-consuming, labor-intensive and expensive [1]. It demands precision and consistency all the time, which is not possible for a human, as human fatigue sets in after a certain period of monotonous activity [2]. Human graders have also been found to suffer from sequential effects during essay grading. Moreover, even though human graders have been rigorously trained, differences in their background,training and their experience in the grading task produce subtle but significant differences in the grading performance [3, 4]. A practical solution to the many problems associated with manual grading, is to have an automated process of essay grading [5-8]. Automated Essay Grading (AEG) can be defined as a computer-based grading system that takes essays as input and assigns suitable grades to the essays, as an output. AEG systems aim to use the ability of the computer - to evaluate and score written prose according to some pre-defined criteria.Studies have shown

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 484–493, 2011.
© Springer-Verlag Berlin Heidelberg 2011

that this automated process is becoming the preferred way of grading, and an increasing number of people are showing an interest in learning about the benefits of using AEG [4, 9-11].

It is common practice to use an essay dataset to build an AEG system. ‘Noisy’ essays in the dataset can potentially hamper and run down the proper performance of an AEG system. So it is important to detect them and eliminate them from the dataset. In this paper, we propose a methodology to reduce noise in the essay dataset. The remainder of this paper is organized as follows. In Section 2, we present various existing approaches to automated essay grading, and their shortcomings. In Section 3, we give an overview of 'OzEgrader' and describe the methodology for reducing noise in the essay dataset. Section 4 presents the results obtained and a discussion of the same is given in Section 5. Section 6 concludes this paper.

2 Existing AEG Approaches

Automated Essay Grading (AEG)

Attempts at automated essay grading started in the 1960s. The earliest attempt was made in 1966 by Page [12] and was called the Project Essay Grade (PEG). This was followed by several approaches. The basic idea of an AEG system is more or less, the same. A substantially large set of prompt-specific essays are pre-scored by human expert graders. This set is divided into two-training set and testing set. The training set is used for developing the scoring model of the AEG system and attuning it. Then this scoring model is used to assign scores to the essays in the testing set. The preformance of the scoring model is typically validated by calculating how well the scoring model “replicated” the scores assigned by the human expert graders.

We have broadly classified the existing AEG approaches into three groups. They are: 1) NLP techniques-based methods, 2) Statistical techniques-based methods, and 3) Hybrid methods. In the next sub-section, we give a brief overview of each of these approaches.

1. NLP Techniques-Based Methods

Natural Language Processing (NLP) is the application of computational methods to analyse characteristics of electronic files of text or speech [13]. All methods that employ language processing techniques such as parsing, pre-processing and lexical-similarity comparisons, can be classified under this category [14]. In the existing literature, NLP techniques-based methods are IEA [15], ETS1 [16], C-Rater [13] and Automark [17].

2. Statistical Techniques-Based Methods

Statistics is the science of making effective use of numerical data relating to groups of individuals or experiments [18]. In recent years, a number of statistical learning methods such as regression models, nearest neighbour classifiers, Bayes belief networks and neural networks have been applied to solve the problem of automated text categorization [19]. In the existing literature, the methods that can be classified as statistical techniques-based methods are PEG [12], SEAR [20], TCT [19], BETSY [11] and IEMS [21].

3. Hybrid Methods

We define 'Hybrid methods' as the methods that are based on a combination of NLP and statistical techniques, thereby capturing most of the advantages of these techniques. In the existing literature, hybrid methods are E-rater [13], PS-ME [22], AEA [5], IntelliMetric [23] and MarkIT [24]. In Table 1, we present a snapshot of the existing AEG methods along with the shortcomings associated with each one of them. For further detailed reading, the reader is directed to the cited references. It is important to note that all the existing AEG systems assume that there exists a linear relationship between the essay features and the essay grade, which is not necessarily true. For example, most of the existing AEG systems assume that the length of the essay is an indication of the quality of the essay. In other words, it is assumed that the longer the essay is, higher the essay grade would be. However, this might not be true in all cases because even though an essay is very long, it might contain irrelevant content. So in cases such as this, the essay grade should be low. In order to address this issue, in section 3, we propose an AEG system that uses intelligent techniques to model both the linear and non-linear relationship between the essay features and the essay grade.

4. Noise Reduction

Futagi [25] stresses the importance of detection of noise in essays, which is an often-ignored topic when discussing automated scoring of essays. He attributes the misspellings, context-based spelling errors, morphological, or syntax-based errors and punctuation errors in the essays to the 'noise'. One of his key findings is that, if an AEG system is built without initially filtering the noise from the essays, then it has a detrimental effect on the performance of the system. It is important to detect noisy essays because an AEG system that has been developed to assess and grade essays would throw unexpected errors if it encountered a figure or a blank response.

Only two existing AEG systems mention some rudimentary techniques for noise detection. My!Access and Criterion, the web-based versions of IntelliMetric and E-rater respectively detect noisy essays of two types [13, 23]. In both the systems, feedback regarding the brevity and off-topicness of the essay is provided.

Table 1. Existing AEG approaches and their shortcomings

AEG system	Technique	Developers	Year	Shortcomings
PEG	Statistical	Page et al	1966	It assesses only the style of writing.
IEA	NLP	Landauer et al	1998	It makes no use of word order.
E-Rater	Hybrid	Burstein et al	1998	It is prone to errors without human intervention.
IntelliMetric	Hybrid	Vantage Learning	1998	No significant studies have been conducted to demonstrate the weaknesses of this system.
TCT	Statistical	Larkey	1998	Structure of the essay is not taken into account.
SEAR	Statistical	J. R. Christie	1998	It assesses content of only technical essays.
ETS(1)	NLP	Burstein et al	1999	It assesses only content.
IEMS	Statistical	Ming et al	2000	No holistic feature vector is defined
C-Rater	NLP	Burstein et al	2001	It gives no regard to writing skills.
BETSY	Statistical	L. M. Rudner	2002	No holistic feature vector is defined
PS-ME	Hybrid	Mason et al	2002	It does not model the non-linear relationship between the essay feature vector and the essay grade
Automark	NLP	Mitchell et al	2002	It does not model the non-linear relationship between the essay feature vector and the essay grade.
AEA	Hybrid	Kakkonen et al	2003	No holistic feature vector is defined.
MarkIT	Hybrid	Williams et al	2004	It is similar to "Bag of words" approach.

3 Methodology and Approach

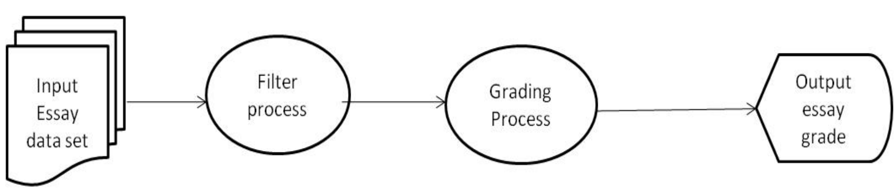


Fig. 1. Overview of OzEgrader

The "OzEgrader" (pronounced "Ozzigrader") is a hybrid method that employs both NLP- and Statistical-based techniques. Overview of our system is illustrated in Fig. 1 and each component of the figure is explained below.

1. Input Essay Dataset

The input dataset consists of essays written by primary, secondary and high school students of Western Australia. Their hand-written essays are word processed, while preserving the errors, and then used as the input essay dataset to OzEgrader. However, before the actual grading process begins, the input essay dataset needs to be scanned through the 'Filter process' in order to filter out essays that are 'noisy'. The filter process is elaborated upon in the following sub-section.

2. Filter Process (Noise Reduction Methodology)

This process is sub-divided into two phases and is diagrammatically represented in Fig. 2. The two phases in Filter process are the Filter Process Stage 1 (FPS 1) and the Filter Process Stage 2 (FPS 2).

a) Filter Process Stage 1

In FPS 1, we are detecting anomalous essays. An 'anomalous' essay can be defined as an essay that has an undesired illustration or characteristic. The different types/cases of anomalous essays and the required action in each case are listed in Table 2. In Fig. 2, in FPS1, the essays that are found to be anomalous are stored in the "Anomalous Essays" dataset, except for Case 5, where the anomalous essay is converted into lower-case letters and added to the output of this phase, "FPS 1 Essay Dataset". At this stage, the FPS 1 Essay Dataset still has 'noisy' essays, called "Poor essays". These are filtered out in the next stage of the filter process.

b) Filter Process Stage 2 (FPS 2)

In FPS 2, we are detecting poor essays. A 'poor' essay can be defined as an essay that is

- gobbledygook (mainly random typing), or
- extremely poor in spelling and punctuation, or
- too small to be called an essay (the essay consists of only a few words or sentence fragments)

According to the Standard American English Dictionary, an ‘essay’ is “a written answer that includes information and discussion, usually to test how well the student understands the subject”. For instance, in a narrative essay, the writer tells a story or part of a story,mainly from the viewpoint of a central character/characters. Also, a narrative essay is comprised of an orientation, complication and a resolution [26]. Hence, a minimum of 75 words is desired in an essay.

Table 2. Anomalous Essays Cases 1 to 5

Case	Type of Anomalous essay	Required Action
1	Blank response (student submits a blank paper)	Scan the essay for null characters and store essay in 'Anomalous essays'
2	Drawing a picture (student draws a picture instead of writing an essay)	Scan the essay for drawings and store essay in 'Anomalous essays'
3	Drawing a picture and writing some words	Scan the essay for drawings and grade the words for 'Spelling'
4	Copying the question prompt as answer - partly or completely	Find similarity between question prompt and answer essay
5	Writing essay completely in upper-case letters	Convert to lower-case letters and add the essay to the output 'FPS1 Essay Dataset'.

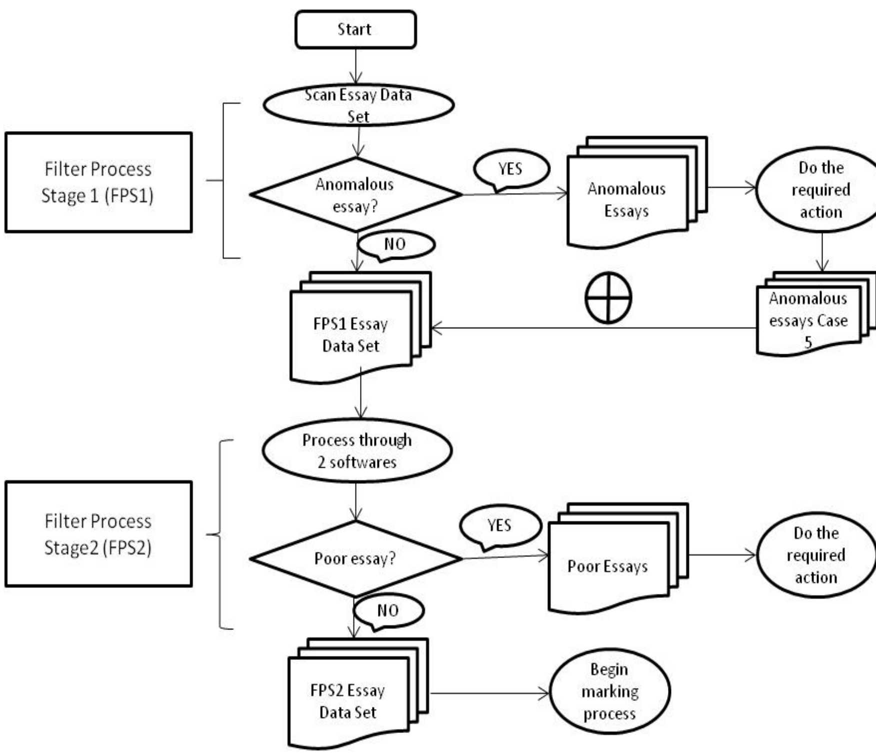


Fig. 2. Conceptual framework of Filter process (Noise reduction methodology)

In Fig. 2, in FPS 2, we scan the FPS 1 Essay Dataset, one essay at a time, through two software programs. We do this in order to automatically obtain various surface features of the essays like number of spelling errors, number of punctuation errors, total number of paragraphs and total number of words in each essay. We use these values to detect poor essays.

Detection of Poor Essays

There are two cases of 'poor' essays; at least one of them needs to be satisfied in order for an essay to be classified as a 'Poor' essay.

Case 1: If (NSE) OR (NPE) $\geq$ 0.1
Then instance/essay = Poor Essay
Poor Essay = TRUE
Where,
(i) NSE = SE/W
NSE = Normalised Spelling Error
SE = Number of Spelling errors
W = Total number of words and
(ii) NPE = PE/W
NPE = Normalised Punctuation Error
PE = Number of Punctuation errors
W = Total number of words
Else
Case 2: If W < 75
Then instance/essay = Poor Essay
Poor Essay = TRUE
where W = Total number of words
Else
instance/essay = FPS2 Essay Data Set
Poor Essay = FALSE.

In Fig. 2, the essays that are classified as 'Poor essays' are stored in the "Poor essays" dataset and the required action is to be taken. All the remaining essays of this phase i.e., the essays that are neither anomalous nor poor, are stored in the "FPS 2 Essay dataset". This is the actual input to the "Grading process", explained in the following sub-section.

c) Grading Process

The grading process is based on the NAPLAN rubric [26], which is a pre-defined rubric for essay grading. This rubric covers all the features of the essay, encompassing every aspect of essay content and style. According to the rubric, the various features of a narrative essay are audience, text structure, character and setting, paragraphing, vocabulary, sentence structure, punctuation, spelling, cohesion and ideas. All these features have a relationship, either linear or non-linear, with the final essay grade. In order to model these relationships, intelligent techniques such as part-of-speech tagging, named entity recognition, artificial neural networks, and fuzzy regression are employed for the grading process to efficiently “model the human brain” [27].

4 Results

All the conditions for anomalous essays and poor essays are mutually exclusive of each other. The noise reduction methodology, explained earlier in Section 3, has been tested on an essay dataset of 313 essays and the results obtained are shown in Table 3.

Table 3. Number of 'noisy' essays detected (out of 313 essays)

<table>
<tr><td rowspan="5">Anomalous Essays</td><td>Case 1-Blank response</td><td>2 essays</td></tr>
<tr><td>Case 2-Drawing a picture</td><td>1 essay</td></tr>
<tr><td>Case 3-Drawing a picture and writing some words</td><td>4 essays</td></tr>
<tr><td>Case 4-Copying question prompt as answer</td><td>2 essays</td></tr>
<tr><td>Case 5-Writing completely in upper-case letters</td><td>3 essays</td></tr>
<tr><td rowspan="2">Poor Essays</td><td>Case 1- Extremely poor in spelling and punctuation</td><td>104 essays</td></tr>
<tr><td>Case 2- Less than 75 words</td><td>10 essays</td></tr>
</table>

Table 3 gives the number of noisy essays detected in each case/category of anomalous and poor essays. Out of a dataset of 313 essays, a total of 12 anomalous essays and 114 poor essays were detected successfully.

5 Discussion

Of the 313 essays dataset, almost all the noisy essays were successfully detected. However, two kinds of essays escaped detection during the filter process: (i) essays where the student used both upper case and lower case letters in an alternating manner, as in "ThEre WaS A dOg". In order to detect this, the condition for anomalous essays, case 5 needs to be adjusted accordingly. (ii) essays that are written in an unconventional way, as an interview or as diary entries. Although these kinds of essays are rare, we still need to detect them in order to grade them properly.

It is important to detect noise in the essay dataset before the actual grading process because noisy essays can adversely affect the performance and efficiency of the grading process.

Despite the fact that the noise reduction methodology has shown very good performance so far, there is still scope for improvement. We acknowledge that the condition 1 for 'Poor essays' is quite high (0.1). It is desirable that the threshold be decreased further to 0.2 or 0.3. Otherwise, there is a problem of having too many poor essays being filtered out. This would happen mostly because students in primary and secondary school are still learning and experimenting with language and are thus bound to have many spelling and punctuation errors. In future, it is recommended that a filter of the filter process be created in order to address this problem.

6 Conclusion

In this paper, we mentioned various existing AEG approaches and the need for OzEgrader. We have explained the methodology and approach for noise reduction. After implementing this methodology, we successfully reduced most of the noise in the essay dataset. The essay dataset is now cleansed and can be used for assessment and evaluation by our AEG system. In future, we plan to further improve the performance of the noise reduction methodology by creating a filter of the filter process, as mentioned earlier. Additionally, we plan to build a robust framework for the implementation of OzEgrader. Consequently, we will develop several algorithms to assess the various criteria of the essay, according to the NAPLAN rubric, for all cases of essays.

Acknowledgements. This work is made possible by the Western Australia Department of Education and Training and with the grant from the Australian Research Council. Special thanks to Ms. Marilyn McKee for her invaluable advice and support in providing the NAPLAN marking rubric and organizing training sessions with other markers to aid in our understanding of the marking process.

References

1. Hardy, R.A.: Examining the cost of performance assessment. Applied Measurement in Education 8, 121–134 (1995)
2. Valenti, S., Neri, F., Cucchiarelli, A.: An overview of current research on automated essay grading. Journal of Information Technology Education 2, 319–330 (2003)
3. Stalnaker, J.M.: The problem of the English examination. Education Record 17, 41 (1936)
4. Rudner, L., Gagne., P.: An overview of three approaches to scoring written essays by computer. Practical Assessment, Research & Evaluation 7(26) (2001)
5. Kakkonen, T.: Computer Assisted Assessment of Essays. Department of Computer Science, Master's. University of Joensuu, Finland (2003)
6. Page, E.B., Petersen, N.S.: The computer moves into essay grading. Phi Delta Kappa 76(7), 561–565 (1995)
7. Page, E.B.: Computer grading of student prose, using modern concepts and software. Journal of Experimental Education 62, 127–142 (1994)
8. Darus, S., Stapa, S., Hussin, S.: Experimenting a Computer-Based Essay Marking System at Universiti Kebangsaan Malaysia. Jurnal Teknologi 39(E), 1–18 (2003)
9. Haswell, R.H.: Automated text-checkers: A chronology and a bibliography of commentary. Computers and Composition Online (2005)
10. Burstein, J.: The e-rater scoring engine: Automated Essay Scoring with natural language processing. In: Shermis, M.D., Burstein, J.C. (eds.) Automated Essay Scoring: A Cross Disciplinary Approach. Lawrence Erlbaum Associates, Mahwah (2003)
11. Rudner, L.M., Liang, T.: Automated essay scoring using Bayes' Theorem. The Journal of Technology, Learning and Assessment 1(2), 3–21 (2002)
12. Page, E.B.: The imminence of grading essays by computer. Phi Delta Kappa 47, 238–243 (1966)

13. Burstein, J., Leacock, C., Swartz, R.: Automated evaluation of essay and short answers. In: Danson, M. (ed.) Sixth International Computer Assisted Assessment Conference. Loughborough University, Loughborough (2001)
14. Kakkonen, T., Myller, N., Suttinen, E., Timonen, J.: Comparison of Dimension Reduction Methods for Automated Essay Grading. Educational Technology & Society 11(3), 275–288 (2008)
15. Landauer, T.K., Laham, D., Rehder, B., Schreiner, M.E.: How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In: Conference How Well Can Passage Meaning Be Derived without Using Word Order? A Comparison of Latent Semantic Analysis and Humans, pp. 412–417. Erlbaum (1991)
16. Whittington, D., Hunt, H.: Approaches to the computerized assessment of free text responses. In: Danson, M. (ed.) Sixth International Computer Assisted Assessment Conference. Loughborough University, UK (1999)
17. Mitchell, T., Russel, T., Broomhead, P., Aldridge, N.: Towards robust computerized marking of free-text responses. In: Danson, M. (ed.) Sixth International Computer Assisted Assessment Conference. Loughborough University, Loughborough (2002)
18. Johnson, R.: Elementary Statistics. PWS-Kent Publishers (1988)
19. Yang, Y.: An evaluation of statistical approaches to text categorization. Carnegie Mellon University (1997)
20. Christie, J.R.: Automated essay marking-for both style and content. In: Danson, M. (ed.) Third Annual Computer Assisted Assessment Conference. Loughborough University, Loughborough (1999)
21. Ming, P.Y., Mikhailov, A.A., Kuan, T.L.: Intelligent Essay Marking System. Learners Together (2000)
22. Mason, O., Stephenson, I.-G.: Automated free text marking with Paperless School. In: Danson, M. (ed.) Sixth International Computer Assisted Assessment Conference. Loughborough University, Loughborough (2002)
23. Dikli, S.: An overview of automated scoring of essays. The Journal of Technology, Learning, and Assessment 5(1) (2006)
24. Williams, R., Dreher, H.: Automatically grading essays with Markit©. Journal of Issues in Informing Science and Information Technology 1, 693–700 (2004)
25. Futagi, Y.: The effects of learner errors on the development of a collocation detection tool. In: Fourth Workshop on Analytics for Noisy Unstructured Text Data, pp. 27–34. ACM, Toronto (2010)
26. Department of Education: Narrative Marking Guide 2010, National Assessment Program-Literacy and Numeracy. Government of Western Australia (2010)
27. Dillon, T.S., Niebur, D.: Neural networks applications in power systems. CRL Publishing Ltd., Market Harborough, UK (1996)

CAREY: ClimAtological ContRol of EmergencY Regions

Maribel Acosta, Marlene Goncalves, and Maria-Esther Vidal

Universidad Simón Bolívar, Venezuela
{macosta,mgoncalves,mvidal}@ldc.usb.ve

Abstract. Nowadays, climate changes are impacting life on Earth; ecological effects as warming of sea-surface temperatures, catastrophic events as storms or mudslides, and the increase of infectious diseases, are affecting life and development. Unfortunately, experts predict that global temperatures will increase even more during the next years; thus, to decide how to assist possibly affected people, experts require tools that help them to discover potential risky regions based on their weather conditions. We address this problem and propose a tool able to support experts in the discovery of these risky areas. We present CAREY, a federated tool built on top of a weather database, that implements a semi-supervised data mining approach to discover regions with similar weather observations which may characterize micro-climate zones. Additionally, Top-k Skyline techniques have been developed to rank micro-climate areas according to how close they are to a given weather condition of risk. We conducted an initial experimental study as a proof-of-concepts, and the preliminary results suggest that CAREY may provide an effective support for the visualization of potential risky areas.

1 Introduction

World climate changes are perturbing the livelihoods of large populations [11]. Sea-surface temperatures have increased, rainfall patterns have changed, and ocean acidification has been altered. Simultaneously, data indicate that frequency of extreme events such that mudslides, droughts and floods, are in rapid increase [6]. Based on correlations between patterns of extreme events and global warming, scientists have linked climate changes to high frequency of natural disasters, and are predicting that these will be far more common in the future [6].

Increasing prevalence of disasters are causing billions of dollars in damage, and are impacting on countries economics [6]. Tools that support experts in the discovery of possibly affected areas based on their weather conditions, may help to rapidly assist affected people. In this paper we present CAREY, a tool for ClimAtological contRol of EmergencY regions, able to support experts in the discovery of potential critical areas based on how best they meet a weather risk condition. CAREY represents sensor observations as RDF properties; the ontology Observation and Measurements in OWL (O&M-OWL) is used to describe the semantics of the observations. Weather observations are grouped according to their Geospatial information then, proximate regions are clustered in terms of the similarity of the climatological conditions into micro-climate areas. Clustering techniques such as k-means, simpleKMeans [13] or X-means [15], are

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 494–503, 2011.
© Springer-Verlag Berlin Heidelberg 2011

used to identify micro-climate areas based on these weather observations. Then, areas are ranked to identify the ones that are possibly in risk. Thus, a risk condition is modeled as a set of thresholds on the values of the sensor data; for example, a risk condition may establish that an area is in risk when the values of humidity are at least 90% and the temperature is at least $100°F$. Skyline and Top-k ranking techniques are used to rank the micro-climate areas and determine the top-k critical ones.

Clustering analysis asset to grouping a set of elements into subsets named clusters, so that, elements in one cluster are similar under given criteria. Top-k Skyline approaches focus on identifying points in the skyline with the highest or best values with respect to a given score criteria. Thus, building on related work, we devise a two-fold solution to the problem of identifying the top-k critical areas that best meet a risk condition. First, during the phase of clustering, Geospatial information is considered to compute proximate regions; altitude, sea level, and Geospatial coordinates are taken into account. Then, sensor observations are used to cluster the regions based on the similarity of their weather observations. Second, micro-climate areas are ranked to identify the ones that belong to the top-k skyline. To efficiently achieve these ranking tasks, CAREY implements a new Top-k Skyline algorithm named TKSIReg, that outperforms state-of-the-art techniques at least by a factor of 2. We empirically studied the quality of the proposed approach, and CAREY has exhibited recall values of up to 85% while precision is close to 75%, suggesting that it may provide an effective and efficient support for the discovery task of risky areas.

This paper is comprised of five additional sections. Section 2 illustrates a motivating example; section 3 summarizes existing state-of-the-art approaches. Section 4 describes the CAREY architecture; section 5 presents an experimental study where we report on the performance of the clustering techniques and the quality of the Top-k Skyline approach. Finally, we conclude in section 6 with an outlook to future work.

2 Motivating Example

Consider a network of weather stations located all around Venezuela; each station measures variables such as temperature, rainfall, etc. Sensor data is used to determine regions with micro-climates, i.e., zones where their weather conditions considerably differ from nearby areas. Micro-climate areas are common in regions with high elevations or close to large bodies of water, where each region can have itself a particular climate which may vary different times in a day.

If an area with a micro-climate groups a set of observations, data should be similar. Based on this partitioning criterion, weather observations can be grouped into clusters where the centroid corresponds to the mean weather measurements per area. Suppose government disaster control agencies are interested in determining critical regions in terms of a risk condition. Suppose only temperature and rainfall are considered, which are both equally important. Further, assume that a region is at risk, if its temperature is less than $10°F$ and its rainfall is greater or equal than 50mm in a day. Thus, a region will be more risky, if there is no other region with a lower temperature and a higher rainfall among the regions with temperature less than $10°F$ and rainfall greater or equal than 50mm.

Formally, a set of more risky regions is comprised of regions that are non-dominated by any other region in terms of the former criteria; this set of regions is called *Skyline*. A region A dominates a region B, if B has greater value in temperature and less in rainfall than A. Based on this, regions 1, 2, 3, and 4, are the non-dominated ones.

Finally, assume that the government agencies will first assist the most critical areas. So, criteria to aggregate the distances between the temperature and rainfall of the micro-climate areas and the risk conditions need to be considered. Table in Figure 1(a) illustrates for each region, the distance between the aggregate weather observations of a region and the risk condition. For example, region 1 is characterized by the average weather values (*Rainfall* 99, *Temperature* 8.4), the Euclidean distance metric to this condition is 49.02.

	Sensor Observations		
Region	**Rainfall**	**Temperature**	**Euclidean Distance**
1	99	8.4	49.02
2	90.2	6	40.67
3	95	6.6	45.12
4	87	1.6	37.94

(a) Euclidean **Distance** to Risk Condition

(b) The Top-3 Critical Regions of Weather Stations

Fig. 1. Risky Micro-climate Areas

Suppose that the three most critical areas will be assisted first, then the Euclidean distance metric can be used as a score function to identify the top-3 areas. Based on this assumption, regions 4, 2 and 3 are the ones that have rainfall and temperature observations that best meet the risk condition. Figure 1(b) illustrates the top-3 critical areas. To select the critical areas, CAREY implements a two-fold approach, and combines cluster analysis and Top-k Skyline techniques to identify the top-k critical regions.

3 Related Work

Data mining techniques have been successfully used to predict patterns between data of different domains. For example, clustering techniques have been studied to summarize the conditions of regions of oceans and atmosphera [17]. Mithal et al. [14] et al. propose data mining techniques to identify patterns that suggest relationships between climate changes and forest degradation. Race et al. [16] relay on clustering techniques to link sea surface temperatures and hurricane frequencies. Finally, Basak et al.[4] propose a machine learning technique to mine atmospheric data and identify patterns that can help to predict future climate conditions.

Skyline [5] and Top-k [7] approaches have been defined in the context of databases to distinguish the best points that satisfy a given ranking condition. A skyline-based technique identifies a partially ordered set of services whose order is induced by criteria comprised of conditions on equally important parameters. Top-k approaches select the top-k elements based on a score function or discriminatory criteria that induces a totally ordered of the input set. Top-k Skyline is a hybrid approach that combines the benefits of Skyline and Top-k [12]. Top-k Skyline identifies the top-k points using discriminatory criteria that induce a total order of the points that compose the skyline and satisfy the multi-dimensional criteria. Top-k Skyline has became necessary in many real-world situations [8], and a variety of ranking metrics have been proposed to discriminate among the points in the skyline, e.g., Skyline Frequency [9] and k-dominant skyline [8]. Goncalves et al. [12] propose an index-based technique called TKSI, to compute the top-k skyline points by just probing the minimal subset of incomparable points and using a given score function to distinguish the best points. Since the properties of the function used to identify the top-k skyline are unknown, the algorithm requires to compute the values of all the points. Similar solutions have been presented by [2,3,10]. However, they completely build the skyline, which can be inefficient. In contrast, the TKSIReg algorithm implemented in CAREY, select the top-k Skyline regions in terms of the Euclidean distance metric; thus, by exploiting the properties of this metric, TKSIReg just requires the computation of the metric for a small subset of points; thus, the performance is improved.

4 Our Approach

4.1 The CAREY Architecture

CAREY is built on top of the Geospatial Web which comprises sensor data annotated with the O&M-OWL ontology which is accessible through a federation of Semantic Sensor Services. CAREY is based on the architecture of wrappers and mediators [19], and integrates data exported by the services (Figure 2). CAREY receives a risk condition that establishes the risk thresholds for each of the sensor observation parameters and the number of risky regions that can be first assisted; the answer of a request is the top-k critical regions that best meet the risk condition.

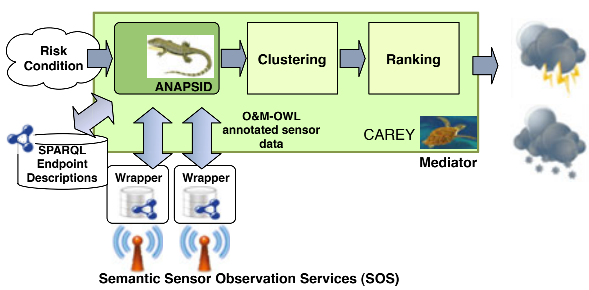


Fig. 2. The CAREY Architecture

ANAPSID [1] query engine is used to recover data from services and integrate all the observations in a unified format. *Wrappers* are created around Semantic Sensor Services to properly build the service URL and to convert the results into appropriate formats. *Mediators* maintain information and statistics about the Web services. CAREY is comprised of two main components:

- *Clustering*: is an out-of-shell component implemented by WEKA[1], that groups sensor data into clusters in two steps. First, Geospatial information is considered to compute proximate regions; altitude, sea level, and Geospatial coordinates are taken into account in this step. Then, sensor observations are used to cluster the regions based on the similarity of their climatological conditions. These clusters represent micro-climate areas or zones that characterize a particular climate that is considerable different from the climate of the surrounding areas. The centroids of the clusters correspond to a vector of the mean values of the sensor observations that represents the weather conditions of the micro-climate area; the sum of squares of the observations to cluster centroid is minimized. Different cluster algorithms are used to create the micro-climate areas, e.g., X-means [15]. O&M-OWL annotations are used to determine distances to the centroids.
- *Ranking*: this component implements ranking techniques to identify the top-k regions that best meet a risk condition among the non-dominated regions. Non-dominated regions are micro-climate areas with at least one observation value in the centroid that is better than the same observation parameter of the centroids of the other areas. These areas have also at least one parameter in the centroid whose value is worse or equal than the value of this parameter in the centroids of the other non-dominated areas. Furthermore, a region is top-k if it is among the k regions with the smallest distance to the risk condition. Semantics encoded in the O&M-OWL ontology is used to determine the distance to the risk condition. We propose the TKSIReg algorithm to solve this ranking problem. TKSIReg exploits the properties of the Euclidean distance metric and identifies among the set of skyline of regions, the ones with the smallest values of this metric. The result of executing TKSIReg is the top-k regions that satisfy a risk condition, and that have the most critical values in the majority of the weather measurements, i.e, the top-k skyline regions. TKSIReg resembles the Top-k SkyIndex (TKSI) algorithm introduced in [12], because in both algorithms the number of skyline points that are computed to produce the top-k skyline is minimized. However, TKSIReg differs from TKSI in exploiting the monotonicity property of the Euclidean distance metric and can produce the risky regions without computing the distance for all the input regions. TKSIReg provides the basis to efficiently identify risky regions.

5 Experimental Study

We empirically analyze, as a proof-of-concepts, the performance and quality of the two-fold approach implemented in CAREY; we report on the effectiveness and efficiency.

Datasets: we considered both synthetic and real-world data. The BIRCH Cluster data generator provided by WEKA was used to generate 20 synthetic datasets, each one comprised of 100 regions. In ten datasets, each region is associated with 1,000 weather observations, while in the other 10 datasets, observations in each region

[1] http://www.cs.waikato.ac.nz/ml/weka/

Table 1. Real-world Weather Datasets

Dataset	Countries	#Observations per Country and Total
DS1	Argentina + Venezuela	21,447+19,631=41,079
DS2	Argentina + Venezuela + Spain	21,447+19,631+46,168=87,247
DS3	Argentina + Venezuela + Spain + Germany	21,447+19,631+46,168+115,417=202,664
DS4	Argentina + Venezuela + Spain + Germany + Canada	21,447+19,631+46,168+115,417+293,615=469,279

vary from 500 to 1,000. Real-world datasets are comprised of sensor weather observations published by WeatherUnderground[2]; Table 1 reports on the size of the downloaded datasets.

Evaluation Metrics: we report on runtime performance, which corresponds to the *elapsed time* produced by the *time* command of the Unix operation system. Experiments were executed on a Linux CentOS machine with an Intel Pentium Core2 Duo 3.0 GHz and 8GB RAM. Quality of CAREY clustering and Top-k Skyline techniques are measured in terms of precision and recall.

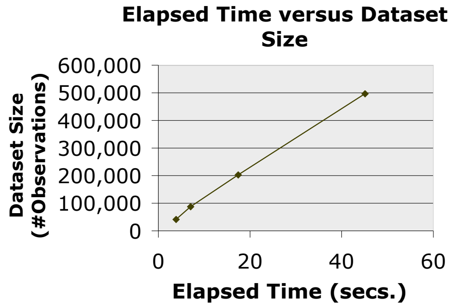
Implementations: CAREY clustering techniques are implemented as the X-means algorithm provided by WEKA. TKSIReg was implemented in Java (64-bit JDK version 1.5.0 12); datasets are stored in Oracle 9i and accessed through the JDBC API.

5.1 Performance and Quality of the CAREY Clustering Techniques

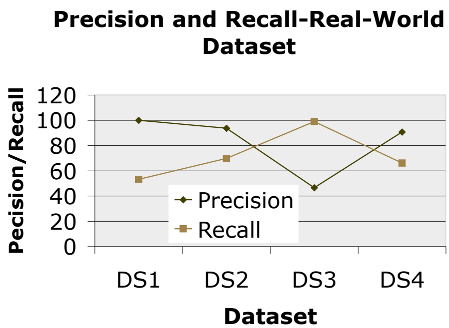
First we report on the performance of the X-means clustering algorithm (Figure 3(a)). We can see that clustering time follows a linear trend with respect to the size of the dataset. Then, we evaluate the quality of CAREY clustering analysis process; we report on precision and recall of the clusters identified by the X-means algorithm, when the real-world datasets in Table 1 were considered. Precision measures the percentage of pairs of observations that were clustered together and belong to the same country; a value of 100% indicates that all the pairs of observations were clustered in their correct country; while a value of 0%, says that all the pairs of observations were clustered incorrectly. Furthermore, recall measures the percentage of pairs of observations that belong to the same country, were assigned in the same cluster. Figure 3(b) reports on the precision and recall for the real datasets DS1, DS2, DS3, and DS4. We can observe that CAREY precisely clusters real weather data; precision and recall are at least of 60%. To better understand the results produced by the clustering algorithm, Figure 4 plots the produced clusters. Analyzing the produced clusters we can observe that even the number of clusters is different to the number of countries in the dataset, stations of the same country that share similar weather observations are usually clustered together.

Finally, we ran X-means to cluster 46,168 temperature observations of Spain for June 2011; Figure 5(a) presents a color map[3], where each of the six colors represents a different range of temperatures. High temperatures are represented with hot colors as red, yellow or orange; low temperatures are described as cold colors, i.e., green and

[2] http://www.wunderground.com/
[3] http://www.opengis.uab.es/



(a) Elapsed Time Versus Dataset Size



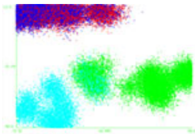
(b) Precision Versus Recall

Fig. 3. Performance and Quality of X-means Real-World Datasets

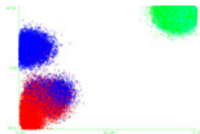
blue. Figure 5(b) reports the clusters produced for this data. We can see that Spain observations are partitioned into four regions. The south and south-west were clustered together and the highest average temperature is $13.3°C$. The north and north-west of Spain, which have a large portion of the territory with low temperatures, were clustered as the region with the coldest temperature, an average of $5.6°C$. Finally, the rest of Spain was clustered in two more regions, one in the east with an average temperature of $8.75°C$, and other in the north-east, with an average of $9.79°C$. Although the number of clusters was sub-estimated, clusters group proximate regions with similar temperatures.

5.2 Quality and Performance of the CAREY Two-Fold Approach

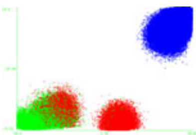
Additionally, we report on the quality and performance for 20 synthetic datasets. Ground truths were computing by running TKSIReg against the real regions of each dataset, which were assigned by the BIRCH Cluster data generator. Precision measures the percentage of regions that belong to the top-k skyline regions computed from the clusters produced by CAREY and belong to the real top-k skyline. A value of 100% indicates that all the top-k skyline regions produced by CAREY are real top-k skyline regions; while a value of 0% says that all the top-k skyline regions produced by



(a) DS1: 2 Countries and 4 Clusters



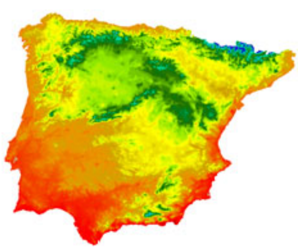
(b) DS2: 3 Countries and 4 Clusters



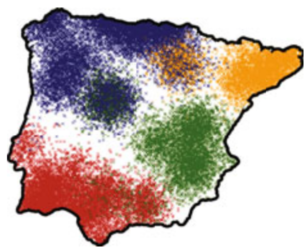
(c) DS3: 4 Countries and 3 Clusters

(d) DS4: 5 Countries and 4 Clusters

Fig. 4. Clusters Produced by X-means for Datasets: DS1, DS2, DS3 and DS4



(a) Weather Observations in Spain. Six microclimates

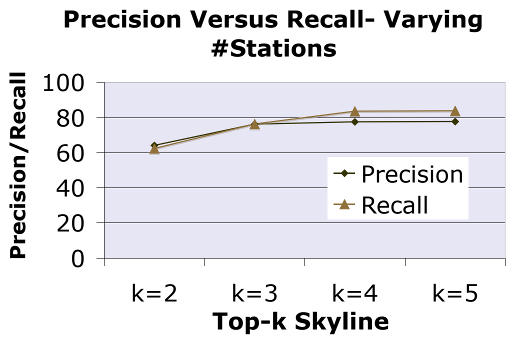


(b) X-means Clustering. Four Clusters

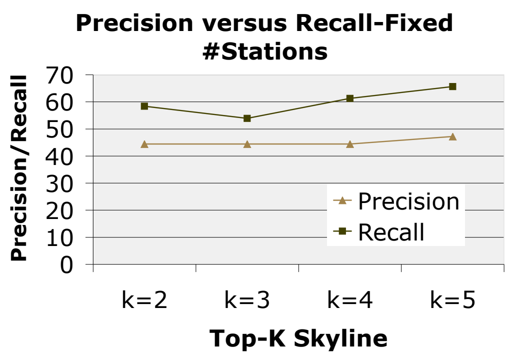
Fig. 5. Real Observations' Distribution versus X-means Clustering

CAREY are not part of the real answer. Furthermore, recall measures the percentage of regions that are part of the real answer and also belong to the top-k skyline regions computed from the produced clusters. Furthermore, we compare the execution time of TKSIReg with respect to time required by TKSI [12] to produce the top-k regions; we can observe that TKSIReg reduces in average the evaluation time of TKSI in 52%.

Figures 6(a) and (b) report on the average precision and recall for values of k from 2 to 5; in each case ten datasets were considered. We can observe that CAREY can better predict the top-k skyline regions if they have a varying number of observations; precision is almost 75% while recall is up to 85%. However, if the regions have a fixed number of observations, CAREY does not perform so well; precision is up to 45% and recall is up to 65%. This may be because the clustering algorithm fails assigning the same number of points to all the clusters. Semantic encoded in the ontology used to describe the weather observations, can be used during clustering analysis to include in a cluster only observations that belong to the same geographical area; however, we notice that regions with the same number of observations are uncommon in real-world.



(a) Regions- Varying # Stations



(b) Regions- Fixed # Stations

Fig. 6. Quality of CAREY Clustering and Ranking Techniques (Average Precision versus Recall). Each point corresponds to the average among ten datasets.

6 Conclusions and Future Work

We have defined CAREY which implements a two-fold approach to identify the top-k critical regions that best meet a risk condition. Regions correspond to clusters of weather observations with similar values. We conducted preliminary empirical study as a proof-of-concept demonstration. Initial experimental results suggest that CAREY techniques may exhibit precision of up to 75% and recall of 85% during the discovery of potential risky areas. In the future we plan to manage more semantics encoded in the data annotations to enhance the clustering and ranking algorithms. Additionally, we will consider other clustering algorithms such as the Shared Near Neighbor approach. Finally, we plan to conduct user studies to define the ground truths to be used during future experiments.

References

1. Acosta, M., Vidal, M.-E., Castillo, J., Lampo, T., Ruckhaus, E.: ANAPSID: AN Adaptive query ProcesSing engIne for sparql enDpoints. In: Accepted at ISWC 2011 (2011)

2. Balke, W.-T., Guntzer, U.: Multi-objective query processing for database systems. In: VLDB, pp. 936–947 (2004)
3. Balke, W.-T., Güntzer, U., Zheng, J.X.: Efficient Distributed Skylining for Web Information Systems. In: Hwang, J., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 256–273. Springer, Heidelberg (2004)
4. Basak, J., Sudarshan, A., Trivedi, D., Santhanam, M.S.: Weather data mining using independent component analysis. Journal of Machine Learning Research 5, 239–253 (2004)
5. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proceedings of the 17th International Conference on Data Engineering, pp. 421–430. IEEE Computer Society, Washington, DC (2001)
6. Carey, J.: Our extreme future: Predicting and coping with the effects of a changing climate. Scientific American 6 (2011)
7. Carey, M.J., Kossmann, D.: On saying "Enough already!" in SQL. SIGMOD Rec. 26(2), 219–230 (1997)
8. Chan, C.-Y., Jagadish, H.V., Tan, K.-L., Tung, A.K.H., Zhang, Z.: Finding k-dominant skylines in high dimensional space. In: SIGMOD 2006: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, pp. 503–514. ACM, New York (2006)
9. Chan, C.-Y., Jagadish, H.V., Tan, K.-L., Tung, A.K.H., Zhang, Z.: On high dimensional skylines. 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. 478–495. Springer, Heidelberg (2006)
10. Chen, L., Gao, S., Anyanwu, K.: Efficiently Evaluating Skyline Queries on RDF Databases. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011. LNCS, vol. 6644, pp. 123–138. Springer, Heidelberg (2011)
11. de Sherbinin, A., Warner, K., Ehrhart, C.: Casualties of climate change: Sea-level rises could displace tens of millions. Scientific American 1 (2011)
12. Goncalves, M., Vidal, M.-E.: Reaching the Top of the Skyline: An Efficient Indexed Algorithm for Top-k Skyline Queries. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 471–485. Springer, Heidelberg (2009)
13. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: The analysis of a simple -means clustering algorithm. In: Symposium on Computational Geometry, pp. 100–109 (2000)
14. Mithal, V., Garg, A., Boriah, S., Steinbach, M., Kumar, V., Potter, C., Klooster, S.A., Castilla-Rubio, J.C.: Monitoring global forest cover using data mining. ACM TIST 2(4), 36 (2011)
15. Pelleg, D., Moore, A.W.: X-means: Extending k-means with efficient estimation of the number of clusters. In: ICML, pp. 727–734 (2000)
16. Race, C., Steinbach, M., Ganguly, A., Semazzi, F.H.M., Kumar, V.: A knowledge discovery strategy for relating sea surface temperatures to frequencies of tropical storms and generating predictions of hurricanes under 21st-century global warming scenarios. In: CIDU, pp. 204–212 (2010)
17. Steinbach, M., Tan, P.-N., Kumar, V., Klooster, S.A., Potter, C.: Discovery of climate indices using clustering. In: KDD, pp. 446–455 (2003)
18. Vlachou, A., Vazirgiannis, M.: Link-based Ranking of Skyline Result Sets. In: Proceedings of the 3rd Multidisciplinary Workshop on Advances in Preference Handling, M-Pref (2007)
19. Wiederhold, G.: Mediators in the architecture of future information systems. IEEE Computer 25(3), 38–49 (1992)

Reasoning with the Depth-First Iterative Deepening Strategy in the DLog System

Adam Meissner and Grażyna Brzykcy

Institute of Control and Information Engineering, Poznań University of Technology, pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland
{Adam.Meissner,Grazyna.Brzykcy}@put.poznan.pl

Abstract. We propose an extension to the DLog system, which is a resolution-based ABox reasoner for the $\mathcal{SHIQ}$ description logic, particularly intended for instance checking and instance retrieval. We replace the original search strategy of the derivation tree, namely the depth-first search, by the depth-first iterative deepening search. The latter approach is proven to be asymptotically optimal among brute-force strategies in terms of proof length, space and time. The extension leads to shorter proofs and, on average, to better timing results when it is enough to compute one positive answer to the given query. We compare the performance of the original and the extended version of the DLog system on a simple benchmark set.

Keywords: DLog reasoner, $\mathcal{SHIQ}$ DL, depth-first iterative deepening search.

1 Introduction

Data processing in contemporary applications is organized around the data sources (i.e. relational databases, RDF repositories). Systems, which contain large amount of underlying data, need functionality that covers powerful but also flexible and efficient data accessing, analysing and integration mechanisms. To effectively cooperate with other heterogeneous data sources the systems also should be augmented with conceptual knowledge. In the W3C standards semantics appears as an ontology, expressed in a variant of the description logic (DL) language [1]. DLs form a wide and rather eclectic class of logical systems generally intended for representing and processing a terminological knowledge. Every DL system contains two basic elements, that is to say *concepts* and *roles*. A concept is a set of *individuals*, which are called *instances* (of the concept). A role, in turn, is a binary relation holding between individuals. A knowledge base expressed in DL is a set of axioms, which can be divided into two disjoint parts, to wit an ABox and a TBox. An ABox consists of *concept assertions* and *role assertions*, namely formulas stating that a given individual is an instance of a given concept or that a pair of individuals belongs to the given role, respectively. A TBox contains axioms describing relationships between concepts and roles, usually in the form of inclusions and equalities. Historically, reasoning

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 504–513, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in DLs is based on tableaux methods and it considers hypothesis referring to a TBox only while an ABox is assumed to be empty. However, contemporary knowledge-based systems intended for ontology processing encompass nonempty ABoxes with a quickly growing number of elements. Hence, for systems of this type, the ABox reasoning becomes more and more important. Furthermore, it comprises two general problems, i.e. *instance checking* and *instance retrieval*. In the first case, a given individual is tested for being an instance of a given concept. In the latter case, the reasoning process is aimed at computing all instances of the given concept.

Unfortunately, pure tableaux methods appeared to be inefficient for ABox processing. A more promising way of solving this problem consists in reducing it to *query answering* in database systems, particularly in deductive databases [13]. A query is a formula, which may contain unbound variables (in short: variables). An *answer* is any mapping of values to variables such that the query becomes a logical consequence of a database. Computing all possible answers to the given query corresponds to instance retrieval task. On the other hand, when a query contains no variables, then the answering process is equivalent to instance checking.

In view of the reasoning engines used in ontology query answering systems, one can differentiate between DL-based and rule-based systems. For example, the Instance Store system [2] with the tableaux algorithm augmented by an efficient access to voluminous ABox belongs to the first category. Among rule-based systems, where the TBox is transformed into rules, diversity of engines and query evaluation strategies is larger. In the famous KAON2 system [5], a DL knowledge base is transformed into an equisatisfiable positive disjunctive Datalog program. In the *DL-Lite* system [15], a resolution-based inference engine is used. Both of the systems are built over a solid logical language and over specialized, efficient reasoning engines but at the expense of language expressivity – $\mathcal{SHIQ}$ DL and $DL\text{-}Lite_A$, respectively.

Another way of designing a knowledge-based system consists in borrowing an existing rule engine as a whole. Examples of suitable software presented in [7] come from Prolog-based systems (XSB, Yap, SWI-Prolog), deductive databases (DLV, IRIS, Ontobroker), production rule systems (Jess used in OWLJessKB and OWL2Jess systems) and rule engines for RDF data (OWLIM systems). A performance and scalability analysis of the engines, presented in [8] by means of results of tests from OpenRuleBench initiative, confirms the leading position of the extended Prolog engine XSB [19], which supports the SGL-WAM [17] with the tabling mechanism. However, it is worth to notice that the evaluation is done only due to data located in the main memory.

Covering the good performance efficiency of a top-down resolution-based inference method in the ontology query answering task, the DLog system has been defined [9] for the full $\mathcal{SHIQ}$ description logic. Performance evaluation shows that DLog is faster than the state-of-the-art description logic tableau-based reasoners, namely RacerPro and Pellet. In many tests the DLog system outperforms also the resolution-based KAON2 system. Because of all the advantages we find the DLog system an interesting and promising tool for knowledge-based query answering problems. We also recognize a possibility of the system performance improvement. For particular reasoning tasks, we suggest to replace the original tree search strategy by a more efficient one, namely by the *depth-first iterative deepening* (DFID) search [6].

We present a new version of the DLog system, where the DFID search is implemented as a metainterpreter. Performance evaluation of the original and the modified reasoner is done on a simple benchmark set. The rest of the paper is organized as follows. Section 2 contains an introduction to reasoning in DLog. The new version of the system is described in section 3. The next section provides and discusses the results of experiments. Section 5 concludes the paper with some final remarks.

2 Principles of the DLog Reasoner

At first, we briefly present the syntax and the semantics of the $\mathcal{SHIQ}$ language. Then, we outline the general idea of query answering in $\mathcal{SHIQ}$ performed by the DLog system.

The $\mathcal{SHIQ}$ language contains two kinds of primary expressions, that is to say *atomic descriptions* (or synonymously, *names*) of concepts and atomic descriptions of roles. Concepts, besides names, can also be represented by complex descriptions, which are built from simpler descriptions and special symbols called *concept constructors*. We use the letter A to denote a concept name and letters C or D as symbols of any concept descriptions. The letter R stands for a role description; the symbol A in the subscript (i.e. R_A) denotes an atomic role description. The set of $\mathcal{SHIQ}$ concept constructors comprises the following elements: *negation* ($\neg C$), *intersection* ($C \sqcap D$), *union* ($C \sqcup D$), *existential restriction* ($\exists R.C$), *value restriction* ($\forall R.C$) and *qualified number restrictions* (($\geq n\ R.C$) and ($\leq n\ R.C$)) – expressions written in parentheses are schemes of relevant concept descriptions. If it does not lead to misunderstanding, in the sequel we often use constructor names to call the descriptions created with them. For example, we say "a negation" instead of "a concept description with a negation as the main constructor". Furthermore, we often identify descriptions with their meanings (e.g. we say "a concept" instead of "a concept description"). All symbols denoting concepts, roles and individuals can possibly be subscripted.

The $\mathcal{SHIQ}$ language also encompasses one role constructor, namely the *inverse* operator (R_A^-), which can be applied to atomic roles only. Moreover, there are two special concepts, to wit $\top$ (*top*) and $\bot$ (*bottom*). The first one denotes the most general concept comprising all considered individuals while the latter represents the least general concept, namely the empty set. TBox axioms can take the form of concept and role inclusions ($\sqsubseteq$) and concept and role equalities ($\equiv$). Additionally, a role can by declared as *transitive* ($\mathsf{Trans}(R)$). ABox, in turn, generally consists of concept assertions ($C(x)$) and role assertions ($R(x, y)$). If an assertion is built of an atomic description, then it is called an *atomic assertion*.

The semantics of the $\mathcal{SHIQ}$ language is defined by means of an interpretation $\mathcal{I}$, which consists of the interpretation domain $\Delta^{\mathcal{I}}$ and the interpretation function $\cdot^{\mathcal{I}}$. The interpretation function assigns a subset of $\Delta^{\mathcal{I}}$ to every concept name and a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to every atomic role. The semantics of other expressions is given in Table 1. For the sake of simplicity, we consider only these elements of the language, which are taken into account in the further discussion. We say that the interpretation $\mathcal{I}$ *satisfies* the description C if it assigns a

Table 1. Some elements of the $\mathcal{SHIQ}$ language

Syntax	Semantics
$\exists R.C$	$\{v \in \Delta^{\mathcal{I}} \mid (\exists w)\left((v,w) \in R^{\mathcal{I}} \wedge w \in C^{\mathcal{I}}\right)\}$
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$C(x)$	$x^{\mathcal{I}} \in C^{\mathcal{I}}$
$R(x,y)$	$(x^{\mathcal{I}}, y^{\mathcal{I}}) \in R^{\mathcal{I}}$

nonempty set to it. Such an interpretation is called a *model* of the concept C. An axiom, in turn, is satisfied by the interpretation $\mathcal{I}$ if the relation stated by the axiom (i.e. inclusion, equality or membership) holds. Furthermore, an interpretation satisfies a knowledge base KB if it satisfies every axiom contained in KB. A concept assertion $C(x)$ is a *logical consequence* of the knowledge base KB (denoted as $KB \models C(x)$) if $C(x)$ is satisfied by every model of KB.

The instance checking problem can be formally defined in terms of logical consequence. Strictly speaking, the *individual x is an instance of the concept C with respect to the knowledge base KB* if and only if $KB \models C(x)$. Moreover, the problem of instance retrieval consists in finding every value x of the variable X occurring in the formula $C(X)$, such that $KB \models C(X/x)$, where the expression X/x denotes the replacement of the variable X by x. In particular case, the instance retrieval can be limited to computing just one value of X.

In query answering systems, the formula $C(x)$ as well as the formula $C(X)$ with the variable X implicitly existentially quantified, is called a *query*. Every value of the variable X representing an instance of the concept C is called an *answer* to the query $C(X)$. Furthermore, we say that an answer to a query is *positive* if and only if the logical consequence relation holds for this query and the given knowledge base. One should observe that in order to show this, it is enough to compute one answer to the query $C(X)$. Otherwise, i.e. when the logical consequence relation does not hold, the answer to the query is *negative*.

In the DLog system, an answer to the query is computed by translating all considered $\mathcal{SHIQ}$ expressions (including the knowledge base and the query) into first-order logic (FOL) formulas. For this purpose, the following conditions have to be complied with: (i) different names are used to denote different individuals (Unique Name Assumption, UNA), (ii) the ABox is consistent and contains only roles, atomic concepts and their negations, (iii) queries are restricted to *conjunctive queries*, to wit they are conjunctions of atomic assertions. One should observe that any assertion of the form $A(x)$ or $R_A(x,y)$ can be interpreted as a first-order atomic formula with the predicate symbol A or R_A, respectively, and with constants x and y as arguments.

The query answering process in DLog is based on the idea of Prolog Technology Theorem Proving (PTTP) [16]. This method consists in proving FOL theorems by translating them into programs in Prolog [14]. Following this approach, the FOL formulas resulting from the initial translation of $\mathcal{SHIQ}$ expressions take the form of *clauses*, to wit they are disjunctions consisting of *positive literals* (i.e. atomic formulas, denoted as L) or *negative literals* (i.e. negated atomic formulas, denoted as $\sim L$). In the next step, every n-element clause is converted into n *contrapositives*. This transformation can be illustrated by the following example. The clause $L_1 \vee L_2 \vee \sim L_3$ is represented by three contrapositives: `L1 :- not_L2, L3.`, `L2 :- not_L1, L3.` and `not_L3 :- not_L1, not_L2.` As one can notice, contrapositives have the syntax of Prolog clauses, however these are not Horn clauses since they may contain a number of predicate symbols with the prefix `not_` denoting negative literals. In order to properly handle symbols of this type, the DLog system (analogously as PTTP) extends the Prolog inference rule by the *model elimination* technique [16]. The extension takes the form of a Prolog code which is added to contrapositives. Furthermore, a numerous optimisations are performed on these formulas, however they are out of scope of this paper (see [9]).

Finally, the resulting query is processed by the Prolog execution environment with regard to the set of contrapositives representing the knowledge base. This process requires only these pieces of data from the ABox, which are needed for query evaluation. In consequence, individuals can be stored in an external database instead of memory. This results in better scalability of the DLog system in comparison to many other ABox reasoners. Strictly speaking, the representation of the knowledge base takes the form of a Prolog program and the query is a Prolog goal. The Prolog execution environment tries to prove the goal by constructing the *derivation tree*. The root of the tree is labelled by the goal, all the other nodes can be obtained from their direct ancestors by one application of the Prolog inference rule. Every branch of the tree ending by an empty goal corresponds to a *successful derivation* (or, synonymously, to a *proof*) and represents an answer to the query. Otherwise, namely when the last node of the branch is labelled by a non-empty goal and no further inference is possible, then the branch represents the *failed derivation*. The tree consisting of failed derivations only is called a *finitely failed derivation tree* and it represents a negative answer.

3 Depth-First Iterative Deepening Search in DLog

PTTP also extends the standard Prolog search strategy, namely the *depth-first search* (DFS) to DFID search. This follows from the fact that DFS is generally incomplete for theorem proving in FOL. Particularly, if the derivation tree contains an infinite branch, the search may get stuck in it, despite of persistence of other successful derivations. The DFID search traverses the tree also in a

DFS manner but only up to the given maximal depth. If this depth has to be exceeded to make another step of derivation, the strategy backtracks. Furthermore, if no proof can be found in the tree up to the current depth limit, then the whole searching process starts again with the maximal depth increased. The initial depth limit, as well as the way it is incremented in every backtrack, are parameters of the presented method. The DFID search is proven to be asymptotically optimal among brute-force strategies with regard to the length of the proof, space and time [6]. In particular, it should be observed that DFID search finds the shortest proof (modulo the depth increment) as the first.

The DLog system, unlike PTTP, uses the standard Prolog search. The DFID strategy is not necessary since conjunctive query answering in $\mathcal{SHIQ}$ is decidable [3]. In consequence, the only infinite branches possibly occurring in the derivation tree come from *cyclic definitions* [1]. Branches of this type can be handled by the *loop elimination* technique, which is less costly in execution, strictly speaking it imposes a lower overhead than the iterative deepening. This argumentation, presented in [9], is right when all answers to the given query are to be returned and thus the whole derivation tree has to be explored. However, the application of the DFID search can essentially speed up the answering process when it is enough to compute one positive answer and, as said in the former paragraph, DFID always finds the shortest proof as the first. This effect may be illustrated by the following example of instance checking. Let us assume a simple knowledge base

$(\exists hasParent.Pole) \sqsubseteq Pole$
$BornInPoland \sqsubseteq Pole$
$hasParent(i_1, i_2)$
$hasParent(i_2, i_3)$
$\ldots$
$hasParent(i_{9999}, i_{10000})$
$BornInPoland(i_1)$
$Pole(i_{10000})$

and the query $Pole(i_1)$. The straightforward Prolog translation of these $\mathcal{SHIQ}$ formulas is as follows

```
pole(X) :- has_parent(X,Y), pole(Y).
pole(X) :- born_in_poland(X).
pole(i10000).

has_parent(i1,i2).
has_parent(i2,i3).
...
has_parent(i9999,i10000).

born_in_poland(i1).

:- pole(i1).
```

One can notice that the standard Prolog search strategy traverses 20000 internal nodes of the derivation tree (i.e. its left-hand side branch) in order to find the proof, since it is computed using the first clause of the predicate `pole/1`. The DFID search, on the other hand, explores just 1 internal node assuming that the initial depth equals 1.

We add the DFID strategy to the Prolog execution environment in the form of a simple metainterpreter presented below. This way of the implementation is less efficient than compiling the DFID search directly into the Prolog code, as it is done in the system PTTP. However, since our intention is to compare two search strategies, the absolute efficiency is not as important as the relative one. Furthermore, the metainterpreter can return negative answers for false queries, which are not taken into account in the system PTTP. We discuss this issue in the sequel.

```
demo(Goal,Depth) :-
   gen_depth_lim(DLim),
   ( call_with_depth_limit(Goal,DLim,Depth) ->
      Depth \== depth_limit_exceeded
    ; !, fail).
```

The predicate `demo/2` takes the `Goal` and executes it using the DFID search. Subsequent values of the depth limit are generated by the `gen_depth_lim/1` predicate. We arbitrarily assume that `DLim` initially equals 1 and it is doubled in every backtrack. The predicate `demo/2` either succeeds or fails whether the `Goal` succeeds or fails respectively, as well. The key element of the metainterpreter is the predicate `call_with_depth_limit/3` available on the SWI Prolog platform. It tries to prove the given `Goal` constructing the derivation tree in a DFS manner but only up to the depth limited by `DLim`. If the proof is found, the predicate succeeds returning the length of the proof as a value of the `Depth` argument. One should consider that this value is guarded by the SWI Prolog execution environment [18] and may differ from the depth following from a theoretical model. If a proof can not be found without exceeding the current depth limit, the predicate also succeeds and binds the constant `depth_limit_exceeded` to the argument `Depth`. Finally, the predicate fails if one can construct a finitely failed derivation tree for `Goal` without exceeding the limited depth.

It should be remarked that the proper handling of finitely failed derivation trees is necessary for the DLog reasoner however it is neglected in the system PTTP. Strictly speaking, PTTP gives the positive answer only if the tested goal is provable, otherwise no answer is returned even if the goal is false. In such a case the system performs an infinite backtrack on the depth generating predicate. This approach is to be accepted for FOL, which is partially decidable, but it is obviously incomplete for decidable logical systems including $\mathcal{SHIQ}$.

4 Experimental Results

We empirically evaluated the efficiency of query answering in the DLog system with DFS and with the DFID search. In both cases, the loop elimination strategy was also included. All the experiments were conducted on a computer equipped with the processor Intel Core i5-680 3.6GHz, 3GB RAM, MS Windows XP Professional ver. 2002 and SWI Prolog 5.8.3. In order to make all results comparable, we used the same metainterpreter to implement both search strategies. The effect of DFS is obtained by assuming an appropriately high value of the initial depth limit, that is to say, a number greater than the height of all processed derivation trees (i.e. 50000).

Analogously to [9], we performed tests on the LUBM ontology (Light University Benchmark) [4]. It is defined in $\mathcal{ALCHI}$ language and describes the domain of the academic world. The ontology was developed particularly for evaluating DL reasoners. In experiments, we considered four variants of LUBM, which are denoted in [9] as `lubm1`, ..., `lubm4`. Each of them consists of the identical TBox part and an ABox with a number of concept assertions and role assertions, which increases for every subsequent variant. The TBox comprises 36 concept inclusions, 6 concept equalities, 5 role inclusions and 4 role equalities. It also contains a transitive role, 21 domain restrictions[1] and 18 range restrictions[1]. The characteristics of all ABoxes is given in [9]. The description of the LUBM ontology also specifies 14 numbered testing queries. However, only 7 of them are suitable for our experiments since the others contain linguistic constructions, which are not supported by the DLog system (e.g. functional roles).

In tests, we considered two evaluation criteria, namely the time (T) of computing the answer and the length (D) of the constructed proof. As stated before, the reasoning process stops after finding the first successful derivation. Results of tests are collected in Table 2. The computational time T is given in seconds and it is taken as an arithmetic mean of three runs. The parameter D, in turn, is regarded as equal to the argument `Depth` of the metainterpreter. We keep the same numeration of queries as in [4]. More precisely, the symbol `q`N in the table corresponds to *Query* N in the mentioned paper, where N denotes the index of the query.

One can observe that the DFS strategy produces generally better timing results than the DFID search when in both cases the constructed derivation has the same length (e.g. `q5/lubm2`, `q13/lubm3`). This follows from the overhead introduced by the iterative deepening. However, it should be noted that in such cases the value of the parameter T for the DFID strategy is still acceptable – usually on the order of a few seconds. In three cases (`q11/lubm4`, `q14/lubm1` and `q14/lubm2`) the computational time for DFS is slightly longer than for DFID search despite of the same value of the parameter D. The probable source of this anomaly is the imprecision of time measurement, which becomes significant for low values of the parameter T (i.e. 0.01s and 0.0s, respectively).

The advantage of the DFID search over DFS occurs when the length of the successful derivation for the former strategy is remarkably smaller than for the latter one. This can be observed for queries `q9/lubm1`, `q9/lubm1` and `q9/lubm4` where the application of the DFID search shortens the computational time even by 1–2 orders of magnitude. In one test, that is to say for the query `q9/lubm3`,

[1] These restrictions are easily expressible in $\mathcal{SHIQ}$; see [1] for guidance.

Table 2. The computational time and the proof length for DFS and DFID search

Query	Strategy	lubm1		lubm2		lubm3		lubm4	
		T	D	T	D	T	D	T	D
q5	DFS	0.02	30	0.09	1357	0.11	1107	0.74	27028
	DFID	0.02	30	0.47	1357	0.66	1107	2.77	27028
q6	DFS	0.01	14	0.02	14	0.03	14	0.03	14
	DFID	0.02	14	0.02	14	0.03	14	0.04	14
q9	DFS	22.36	1631	225.32	3732	0.07	5452	465.11	7751
	DFID	8.82	129	8.74	65	8.69	65	42.31	129
q11	DFS	0	12	0	12	0	33	0.01	183
	DFID	0	12	0	12	0	33	0	183
q12	DFS	0	12	0	30	0	54	0	75
	DFID	0	12	0	17	0	17	0.01	33
q13	DFS	0.03	30	0.09	1359	0.11	1109	0.21	3571
	DFID	0.03	30	0.48	1359	0.68	1109	1.2	3571
q14	DFS	0.01	11	0.01	11	0.01	11	0.01	11
	DFID	0	11	0	11	0.01	11	0.01	11

the value of the parameter T is lower for DFS despite of the fact that the derivation is significantly longer than for DFID search. The possible reason of this effect may be a high branching factor of the derivation tree. This issue, however, requires more thorough observations in future.

5 Final Remarks

We have considered the realisation of an asymptotically optimal DFID search strategy in the DLog system, an ABox reasoner for $\mathcal{SHIQ}$ knowledge bases. In order to answer to a query, the system initially transforms the TBox axioms into Prolog clauses. Then, it applies a top-down resolution-based inference method over the clauses and over a possibly independent data source containing the ABox assertions. The modified strategy is implemented as a metainterpreter. To guarantee an appropriate reliability of comparison experiments, the metainterpreter subsumes the original and the modified search strategy. Tests performed on the Light University Benchmark ontology show that the DFID strategy is more efficient for the certain class of queries, when the derivation for this strategy is shorter than for the standard one (i.e. DFS). However, the comprehensive efficiency analysis of our proposition requires more tests, which are intended for the future. Moreover, due to complexity of a query class determination it seems reasonable to separate the object knowledge and the control knowledge in the system. With a set of different strategies, containing also the parallel and distribute ones (see e.g. [10,11,12]), one can compose the specialized reasoner for the particular class of queries. This work is also planned for future.

References

1. Baader, F., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, implementation, and applications. Cambridge University Press, Cambridge (2003)
2. Bechhofer, S., Horrocks, I., Turi, D.: The OWL Instance Store: System Description. In: Nieuwenhuis, R. (ed.) CADE 2005. LNCS (LNAI), vol. 3632, pp. 177–181. Springer, Heidelberg (2005)
3. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive Query Answering for the Description Logic $\mathcal{SHIQ}$. In: Veloso, M.M. (ed.) IJCAI 2007, pp. 399–404 (2007)
4. Guo, Y., Pan, Z., Heflin, J.: An Evaluation of Knowledge Base Systems for Large OWL Datasets. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 274–288. Springer, Heidelberg (2004)
5. Hustadt, U., Motik, B., Sattler, U.: Data Complexity in Very Expressive Description Logics. In: Proc. of the Nineteenth International Joint Conference on Artificial Intelligence, pp. 466–471 (2005)
6. Korf, R.E.: Depth-first iterative-deepening: an optimal admissible tree search. Artificial Intelligence 27, 97–109 (1985)
7. Liang, S., Fodor, P., Wan, H., Kifer, M.: OpenRuleBench: An Analysis of the Performance of Rule Engines. In: Proc. of the 18th International Conference on World Wide Web, pp. 601–610. ACM, New York (2009)
8. Liang, S., Fodor, P., Wan, H., Kifer, M.: OpenRuleBench: Detailed Report. Department of Computer Science State University of New York, May 29 (2009)
9. Lukácsy, G., Szeredi, P.: Efficient description logic reasoning in Prolog: The DLog system. Theory Pract. Logic Program. 9(3), 343–414 (2009)
10. Meissner, A.: Introducing Parallel Computations to a PTTP-based First-Order Reasoning Process in the Oz Language. In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) SETN 2008. LNCS (LNAI), vol. 5138, pp. 359–364. Springer, Heidelberg (2008)
11. Meissner, A.: A Simple Parallel Reasoning System for the $\mathcal{ALC}$ Description Logic. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS (LNAI), vol. 5796, pp. 413–424. Springer, Heidelberg (2009)
12. Meissner, A., Brzykcy, G.: A Parallel Deduction for Description Logics with $\mathcal{ALC}$ Language. In: Cotta, C., et al. (eds.) Knowledge-Driven Computing. SCI, vol. 102, pp. 149–164. Springer, Heidelberg (2008)
13. Motik, B.: Reasoning in Description Logics using Resolution and Deductive Databases, PhD thesis. University of Karlsruhe (2006)
14. Nilsson, U., Małuszyński, J.: Logic, Programming and Prolog, 2nd edn. John Wiley & Sons Ltd. (1995)
15. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking Data to Ontologies. In: Spaccapietra, S. (ed.) Journal on Data Semantics X. LNCS, vol. 4900, pp. 133–173. Springer, Heidelberg (2008)
16. Stickel, M.: A Prolog Technology Theorem Prover: A New Exposition and Implementation in Prolog. SRI Int. Menlo Park. Technical Note 464 (1989)
17. Swift, T., Warren, D.: An abstract machine for SLG resolution: Definite programs. In: Proc. of the International Symposium on Logic Programming. MIT Press, Cambridge (1994)
18. Wielemaker, J.: SWI-Prolog 5.8 Reference Manual. Univ. of Amsterdam (2010)
19. XSB System, http://xsb.sourceforge.net/

Model-Driven Approach to XML Schema Evolution*

Martin Nečaský, Irena Mlýnková and Jakub Klímek

XML Research Group, Department of Software Engineering
Charles University in Prague, Czech Republic
{necasky,mlynkova,klimek}@ksi.mff.cuni.cz

Abstract. Today, XML is a standard meta-language for representation of exchanged messages between information systems. To enable exchange, the structure of the messages must be established in a form of XML schemas. Usually, more than one type of messages is exchanged and, hence, a family of XML schemas needs to be created. An important task for the designer is, therefore, to design the XML schemas and then evolve them continuously as user requirements change. Doing this manually may be very difficult due to the fact that single change in the user requirements may impact many XML schemas.

In this paper, we present a novel approach to evolution of families of XML schemas. It is based on modeling XML schemas at two levels – conceptual and XML schema. The designer performs a change only once in the conceptual schema and our introduced mechanism propagates the change to all affected XML schemas. Propagation from the XML schema to the conceptual level is also supported.

1 Introduction

Today, XML is a standard metalanguage for representation of messages exchanged between information systems. It is however necessary that communicating parties agree on the structure of exchanged messages. The agreed structure is described as an XML schema expressed in a suitable XML schema language. Usually, more than one type of messages is exchanged among the parties and, hence, a family of XML schemas (sometimes also called *XML vocabulary*) needs to be developed in practice. Examples of such families are open XML vocabularies *OpenTravel* [2], *HL7* [1], and many others.

In our previous work [9,10], we aimed at the problem of designing XML schemas of XML vocabularies. The introduced technique exploits the fact that a vocabulary is usually related to a common data domain, e.g. traveling, health care or public procurement. Therefore, a conceptual schema of the domain is firstly designed. Each XML schema is then modeled as a specific view of the conceptual schema. In this work, we aim at the problem of evolving the XML schemas as user requirements change. A new user requirement may have an impact on several XML schemas in the vocabulary. The designer, therefore, has to identify the impacted XML schemas and determine how they must be evolved. It is a widespread practice today to deal with this task at the level of separate XML schemas expressed in an XML schema language such as DTD or

* Supported in part by the Czech Science Foundation (GAČR), grants number P202/10/0573 and P202/11/P455 and in part by grant SVV-2011-263312.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 514–523, 2011.
© Springer-Verlag Berlin Heidelberg 2011

XML Schema. However, this may be very difficult in case of complex XML vocabularies. Also manual identification of the affected parts is not easy in case of tens or even hundreds of schemas. In this work we introduce a technique based on describing the required changes at the common conceptual level and semi-automatic propagation of the changes to the affected XML schemas.

Related Work. In [16] the authors describe two schema evolution approaches – *incremental* and *change-based*. The former is based on change operations which allow to continuously evolve the schema. The latter is based on comparison of two given versions of the schema. Regarding XML schema evolution, most of the existing works introduce incremental techniques. There are techniques which consider editing XML schemas and differ in the selected XML schema language, i.e. DTD [14,3] or XML Schema [15,7]. Other techniques consider an abstraction of a single XML schema – either visualization [5] or a kind of UML diagram [4]. However, none of them considers a joint evolution of a family of related XML schemas. In [13] multiple *local* XML schemas are considered and mapped to a *global* object-oriented schema. Possible operations with a local schema and their propagation to the global schema are discussed. However, the global schema does not represent a common problem domain, but a common integrated schema, the changes are propagated just from local schemas to the global one and the operations are not defined rigorously. This work is based on our previous results in conceptual modeling for XML. We provide an extensive survey of these approaches in [11]. Their problem is that they understand a conceptual schema as a conceptualization of a single XML schema. They provide methods which automatically translate the conceptual schema to an XML schema. This is useful for deriving an XML database schema but not when designing and evolving a family of XML schemas.

Contribution. We present a novel incremental approach to evolution of a family of XML schemas. First, we introduce a set of operations for evolution of XML schemas at two levels – conceptual and XML schema. We also show that kinds of operations considered by the current literature are not sufficient and we fill in this gap. Second, we introduce a novel mechanism of propagation of operations between both levels which significantly reduces a manual work of a designer when managing a set of XML schemas in a system. We demonstrate the contributions on two real-world case studies.

Outline. The paper is structured as follows. Section 2 is motivating. In Section 3, we introduce the conceptual modeling language for XML. In Section 4, we introduce atomic operations for schema evolution. In Section 5, we introduce the propagation mechanism. In Section 6, we evaluate the introduced approach. Section 7 concludes.

2 Motivation

Let us discuss the evolution problem on two scenarios. First, a designer creates a new family of XML schemas which has not been deployed in a run-time environment yet. He iterates in several iterations before an acceptable version of the XML schemas is prepared to be deployed. He needs a mechanism which shows an impact of each change to the unfinished XML schemas and helps to propagate the change to the XML schemas. Because the XML schemas have not been deployed yet, there are no XML documents. Therefore, it is not necessary to propagate the changes to the XML document level. The second scenario is *adapting an existing family of XML schemas* which have already been deployed in a run-time environment. In this scenario it is necessary to consider XML documents as well. Such scenario usually occurs when new or changed requirements need to be implemented in the running system (e.g. a change in legislation). We have already proposed a technique for propagation of evolution changes from XML schemas to XML documents using XSLT scripts generated using the approach described in this paper in [6]. In this paper, we aim at propagation of changes between the conceptual schema and XML schemas bounded to that conceptual schema. For this, we consider a set of atomic operations which are incrementally applied by the designer to the schemas and appropriately propagated by our mechanism between the conceptual and XML schema levels. The current literature considers addition, removal, migratory, and sedentary operations for schema evolution. However, they are not sufficient, because they do not keep the semantic relationships between existing and newly created schema components. We demonstrate the problem on an example. Let us have a class `Customer`. Its attributes `line1` and `line2` represent a customer's address. Later, the users require a precise differentiation of street, city and country. Therefore, the designer creates new respective attributes `street`, `city` and `country`, and removes the old ones. It is clear that the semantic relationship between the old and new attributes is lost when using only the creation and removal operations. It results in loosing of the respective data that should be transformed accordingly. Therefore, we need an operation which enables to explicitly specify the semantic equivalence between two sets of schema components. There are two possible ways of doing that. The simpler one is to denote that the sets are semantically equivalent without specifying how. The more complex way is to moreover describe the equivalence at the data level in a form of a query expression in a suitable language. The second approach is necessary when adapting XML documents. Since we are now interested only in adapting XML schemas, we adopt the first approach and we introduce a new kind of operations called *synchronizing operations*.

3 Language for Designing XML Formats

In this section, we briefly describe our language for designing XML schemas. Its full description may be found in [10]. The design proceeds at two levels. The *conceptual level* contains a conceptual schema of the problem domain. The *logical level* contains schemas which model the aimed XML schemas in a pictorial and user-friendly way. With respect to the established terminology of Model-Driven Architecture (MDA) [8] we call the two modeling languages *Platform-Independent Model (PIM)* and *Platform-Specific Model (PSM)*, respectively.

3.1 Platform-Independent Model (PIM)

As *platform-independent model (PIM)* we employ the model of UML class diagrams.

Definition 1. *A* PIM schema *is a* 6*-tuple* $\mathcal{S} = (\mathcal{S}_c, \mathcal{S}_a, \mathcal{S}_r, \mathcal{S}_e, class, participant)$. $\mathcal{S}_c$, $\mathcal{S}_a$, *and* $\mathcal{S}_r$ *are sets of* classes, attributes, *and* associations *respectively.* $\mathcal{S}_e$ *is a set of* association ends *in* $\mathcal{S}$. *An association is a set* $R = \{E_1, E_2\}$, *where* E_1, E_2 *are two different association ends. Any two associations are disjoint. Functions* $class$ *and* $participant$ *assign a class to each attribute and association end, respectively.*

We will use $attributes(C)$ and $associations(C)$ to denote the set of all attributes of class C or associations connected to C, respectively. $associations(C_1, C_2)$ will denote all associations connecting two classes C_1 and C_2. For an association $R = \{E_1, E_2\}$, R^{E_1} will denote an ordered image of R. It is an ordered pair (E_1, E_2). We will also use $\overrightarrow{\mathcal{S}_r}$ to denote the set of ordered images of all associations in $\mathcal{S}_r$. A sample PIM schema is depicted in Fig. 1. Note that we also consider names, types and cardinalities of the components in the full version of our formalism [10]. However, we omit them from the definition for simplicity and use them intuitively in the further text.

3.2 Platform-Specific Model (PSM)

The *platform-specific model (PSM)* enables to specify how a part of the reality is represented in a particular XML schema in a UML-style way. We view a PSM schema in two perspectives. From the *grammatical perspective*, it models an XML schema. From the *conceptual perspective*, it delimits the represented part of the reality. Its advantage is clear – the designer works in a UML-style way which is more comfortable than editing the XML schema. Moreover, he may derive the PSM schema from the PIM schema which is easier then designing the respective XML schema manually from scratch.

Definition 2. *A* PSM schema *is an* 8*-tuple* $\mathcal{S}' = (\mathcal{S}'_c, \mathcal{S}'_a, \mathcal{S}'_r, \mathcal{S}'_e, C'_{\mathcal{S}'}, content', class', participant')$. $\mathcal{S}'_c$, $\mathcal{S}'_a$, *and* $\mathcal{S}'_r$ *are sets of* classes, attributes, *and* associations, *respectively.* $\mathcal{S}'_e$ *is a set of* association ends. *An association is an ordered pair* $R' = (E'_1, E'_2)$, *where* E'_1, E'_2 *are different association ends. Any two associations are disjoint.* $C'_{\mathcal{S}'} \in \mathcal{S}'_c$ *is a* schema class *of* $\mathcal{S}'$. *Function* $content'$ *assigns a class* C' *with an ordered sequence of all associations with* C' *as the parent.*

A PSM schema is displayed as a UML class diagram in an ordered tree layout which reflects the hierarchical structure of XML data. For an association $R' = (E'_1, E'_2)$ we call $participant'(E'_1)$ and $participant'(E'_2)$ *parent* and *child* of R' and denote them $parent'(R')$ and $child'(R')$, respectively. We use $attributes'(C')$ to denote the set of attributes of a class C'. Note that, again, we omit names, types and cardinalities from the definition. Apparently we also do not cover all the XML Schema/DTD constructs – they are covered in the full definition of our model [9]. A sample PSM schema is depicted in Fig. 2. An actual XML schema (e.g. XSD or DTD) can be automatically generated from our PSM schema, so when the conceptual model changes, all the user needs to do is simply to generate the XML schemas from the updated conceptual model. Also, our PSM schema can be automatically reverse-engineered from an XML schema. From the *grammatical perspective*, a PSM schema models a particular XML format. A class C' models a group of XML elements and attributes which are modeled by its attributes and associations in its content. An association R' with a name l models an XML element which encloses the group of XML attributes and elements modeled by its child. If R' does not have a name, it just specifies that the group is a part of the group modeled by the parent. An attribute A' models an XML element or attribute depending on its *XML form*. An XML form is visualized before the attribute with '-' (XML element) or '@' (XML attribute) symbol. A sample XML document whose schema is modeled by the sample PSM schema is depicted in Fig. 2. From the *conceptual perspective*, a PSM schema is mapped to a part of a PIM schema which specifies the part of the

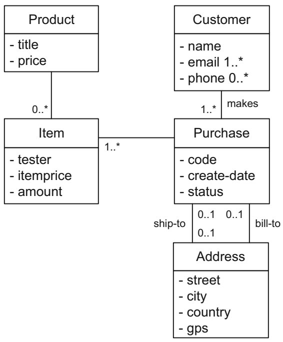
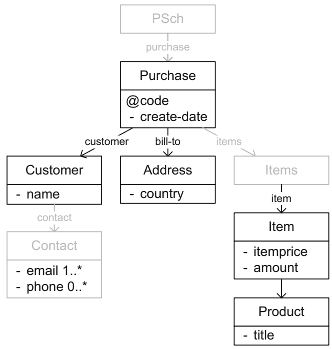


Fig. 1. Sample PIM schema




```
<?xml version="1.0"?>
<purchase code="1234">
  <create-date>9/13/2010</create-date>
  <customer>
    <name>Martin Necasky</name>
    <contact>
      <email>necasky@...</email>
      <phone>+420254...</phone>
      <phone>+420777...</phone>
    </contact>
  </customer>
  <bill-to>
    <country>Czech Republic</country>
  </bill-to>
  <items>
    <item>
      <itemprice>10</itemprice>
      <amount>3</amount>
      <title>White book</title>
    </item>
    <item>
      <itemprice>55</itemprice>
      <amount>1</amount>
      <title>Green book</title>
    </item>
  </items>
</purchase>
```

Fig. 2. Sample PSM schema

reality represented by the modeled XML schema. This mapping is called *interpretation of the PSM schema against the PIM schema*. The interpretation can not be an arbitrary mapping. It must meet certain conditions introduced by the following definition.

Definition 3. *An* interpretation *of a PSM schema $\mathcal{S}'$ against a PIM schema $\mathcal{S}$ is a partial function $I : (\mathcal{S}'_c \rightarrow \mathcal{S}_c) \cup (\mathcal{S}'_a \rightarrow \mathcal{S}_a) \cup (\mathcal{S}'_r \rightarrow \overrightarrow{\mathcal{S}_r})$. For $X' \in (\mathcal{S}'_c \cup \mathcal{S}'_a \cup \mathcal{S}'_r)$, we call $I(X')$* interpretation of *X'. $I(X') = \lambda$ denotes that X' does not have an interpretation. Let $intclass'(X')$ denote the closest ancestor class C' with $I(C') \neq \lambda$ to a component X' (C' is called* interpreted class context *of X', $intclass'(X') = \lambda$ denotes that C' does not exist). The following must be satisfied:*

$$(\forall A' \in \mathcal{S}'_a \text{ s.t. } I(A') \neq \lambda)(class(I(A')) = I(intclass'(A'))) \tag{1}$$

$$(\forall R' \in \mathcal{S}'_r \text{ s.t. } I(child'(R')) \neq \lambda)(I(R') = (E_1, E_2) \wedge \\ participant(E_1) = I(intclass'(R')) \wedge participant(E_2) = I(child'(R'))) \tag{2}$$

$$(\forall R' \in \mathcal{S}'_r)(I(child'(R')) = \lambda \leftrightarrow I(R') = \lambda) \tag{3}$$

The conditions ensure that the semantics of the PSM schema is unambiguously described by I in terms of the PIM schema. There are also components without an interpretation which means that it has no semantic equivalent in the PIM schema. Note that interpretations are not necessarily so intuitive in real-life as they are on our examples. For details, we refer to [10].

4 Operations

In this section, we introduce atomic operations for PIM and PSM schema evolution. We provide their examples and describe the most interesting ones. Note that the atomic operations serve as a formal basis for creating more user-friendly operations composed of the atomic ones. Full details of all atomic operations and how composite operations can be constructed can be found in [12]. In the next section, we describe how atomic operations performed at one level (PIM or PSM) are propagated to the atomic operations at the other level (PSM or PIM, respectively). Any operation composed of the atomic ones is automatically propagated as a sequence of corresponding atomic operations.

Atomic operations for PIM schema evolution provide a formal way of changing the PIM schema. Their examples (affecting classes and attributes) are listed in Tab. 1. There are similar operations for associations. The definitions contain preconditions for some operations (**p:**). An addition operation creates a new component and sets its name, type and cardinality to default values configured by the designer. A sedentary operation updates a name, data type, etc. The precondition of the class removal operation ensures that a class is not removed when it has attributes or connected associations.

Table 1. Examples of atomic operations for PIM schema adaptation

Operation	Kind	Effect
$C = \alpha_c()$	Addition	Adds a new class C.
$A = \alpha_a(C)$	Addition	Adds a new attribute A to an existing class C.
$\delta_c(C)$	Removal	Removes an existing class C. **p:** $attributes(C) = \emptyset \wedge associations(C) = \emptyset$
$\delta_a(A)$	Removal	Removes an existing attribute A.
$\upsilon_a^{name\mid type\mid card}(A, v)$	Sedentary	Updates the name, type, or cardinality, respectively of an attribute A to a new value v.
$\upsilon_a^{class}(A, C_v)$	Migratory	Moves an attribute A to a class C_v. **p:** $associations(class(A), C_v) \neq \emptyset$
$\sigma_a(\mathcal{X}_1, \mathcal{X}_2)$	Synchronizing	Synchronizes two sets of attributes $\mathcal{X}_1$ and $\mathcal{X}_2$. **p:** $(\exists C \in \mathcal{S}_c)(\mathcal{X}_1, \mathcal{X}_2 \subseteq attributes(C))$

The migratory operation $\upsilon_a^{class}(A, C_v)$ allows for moving an attribute . Its precondition requires that there must be an association connecting the current class C_u of A with C_v. This is natural since when we move an attribute, there is typically some semantic relationship between C_u and C_v. This relationship may be modeled in the PIM schema by an association connecting C_u and C_v. It may also be modeled by a path of associations. Or, it can even be not modeled in the PIM schema at all and, instead, only considered by the designer implicitly in his/her mind. Note that the atomic operation only considers the former case. The other cases may be implemented by composing the creation, removal and migratory operations. See [12] for details. The synchronizing operation $\sigma_a(\mathcal{X}_1, \mathcal{X}_2)$ allow for denoting that two sets of attributes are semantically equivalent. They have no direct effect on the structure of the edited PIM schema. The precondition of this operation requires the synchronized attributes to be within the same class. For an example of how these operations can cover changes, let us get back to our motivational example, where we change a representation of a part of customer information from attributes `name`, `line1` and `line2` to attributes `name`, `street`, `city` and `country`. First of all, we add the new attributes: $A_{\texttt{street}} = \alpha_a(\texttt{Customer})$, $A_{\texttt{city}} = \alpha_a(\texttt{Customer})$, $A_{\texttt{country}} = \alpha_a(\texttt{Customer})$. Next, we specify the semantic equivalence of the new and the old set of attributes: $\sigma_a(\{A_{\texttt{name}}, A_{\texttt{line1}}, A_{\texttt{line2}}\}, \{A_{\texttt{name}}, A_{\texttt{street}}, A_{\texttt{city}}, A_{\texttt{country}}\})$. Finally, we remove the old attributes: $\delta_a(A_{\texttt{line1}}), \delta_a(A_{\texttt{line2}})$.

Table 2. Examples of atomic operations for PSM schema adaptation

Operation	Kind	Effect
$C' = \alpha'_c()$	Addition	Adds a new class C'.
$R' = \alpha'_r(C'_1, C'_2)$	Addition	Adds a new association R' going from an existing class C'_1 to another existing class C'_2. **p:** $(\forall R' \in \mathcal{S}'_r)(child'(R') \neq C'_2)$.
$\delta'_c(C')$	Removal	Removes an existing class C' from $\mathcal{S}'$. **p:** $attributes'(C') = \emptyset \wedge associations'(C') = \emptyset$
$\delta'_r(R')$	Removal	Removes an existing association R'.
$\upsilon_r^{name'\|card'}(R', v)$	Sedentary	Updates the name or cardinality of an association R' to a new value v.
$\upsilon_{c\|a\|r}^{int'}(X', X)$	Sedentary	Updates an interpretation of a class, attribute, or association X' to a class, attribute or association X in the PIM schema, respectively.
$\upsilon_r^{participant'}(E', C'_v)$	Migratory	Reconnects an endpoint E' to a class C'_v. **p:** $participant'(E) = parentclass'(C'_v) \vee C'_v = parentclass'(participant'(E'))$
$\sigma'_r(\mathcal{X}'_1, \mathcal{X}'_2)$	Synchronizing	Synchronizes two sets of associations $\mathcal{X}'_1$ and $\mathcal{X}'_2$. **p:** $(\exists C'_1 \in S'_c, C_2 \in S_c)(\forall \mathcal{X}'_1 \cup \mathcal{X}'_2)$ $(parent'(R') = C'_1 \vee child'(R') = C'_2)$

Atomic operations for PSM schema evolution are similar to the previous ones. Their examples are listed in Tab. 2. Only the synchronization of two sets of associations is more complicated. We cannot require that the associations connect the same classes because the PSM schema is a forest of trees. Instead, we require that the associations have only one class in common. The other classes must have the same PIM class as their interpretation.

5 Propagation of Atomic Operations

An interpretation I of a PSM schema $\mathcal{S}'$ against a PIM schema $\mathcal{S}$ must satisfy the conditions given by Def. 3. When $\mathcal{S}$ or $\mathcal{S}'$ is modified, the conditions may be violated and the other must be modified accordingly. We call this process *propagation*. When an atomic operation is executed on $\mathcal{S}$, it must be propagated to all PSM schemas with an interpretation against $\mathcal{S}$. Vice versa, when an atomic operation is executed on $\mathcal{S}'$, it must be propagated to $\mathcal{S}$ and, from here, to the other PSM schemas. This process is called *joint adaptation* of multiple XML schemas related by the PIM schema. Here we discuss only the most interesting parts of propagation mechanism. See [12] for details.

The propagation works for the addition, removal and sedentary operations as follows: An addition is not propagated by our mechanism, because a created component models a new part of reality which is not represented in the other schemas. The removal operations are propagated from the PIM to the PSM level. Removal of a PIM component X has an impact on each PSM component X' s.t. $I(X') = X$. There are two propagation options: X' may be removed as well or $I(X')$ is set to λ.The removal operations in the other direction are not propagated because, according to Def. 3, the existence of a

PIM component does not depend on any PSM component. The sedentary operations are propagated straightforwardly. Types and cardinalities are propagated automatically due to type and cardinality compatibility. Names are propagated only optionally. Finally, the migratory operations propagation must move the interpreted attributes and associations to the PSM schema counterparts of their new PIM classes. This may require creating the PSM classes where they were not before.

Propagation of Synchronizing Operations. Synchronizing of a set $\mathcal{X}_2$ with a set $\mathcal{X}_1$ does not violate Def. 3. But, our propagation mechanism ensures that the equivalence is preserved in PSM schemas. An existence of an equivalent to $\mathcal{X}_1$ implies an existence of an equivalent to $\mathcal{X}_2$ in a given PSM schema, and vice versa. We show its propagation only from PIM to PSM. The reversed propagation is only a technical modification. To demonstrate propagation of *attribute synchronization* suppose a PIM schema in Fig. 3 (a) and two PSM schemas in Fig. 3 (b,d) with interpretations against the PIM schema. Suppose that the designer needs to replace attributes `line1`, `line2` with attributes `street`, `city`, `country` in the PIM schema. One part of this operation is the synchronization of set {`street`, `city`, `country`} with set {`line1`, `line2`}. It means that whenever there are attributes with interpretations `line1` and `line2` in the same interpreted class context C' there must also be attributes with interpretations `street`, `city` and `country` in the same interpreted class context, and vice versa.

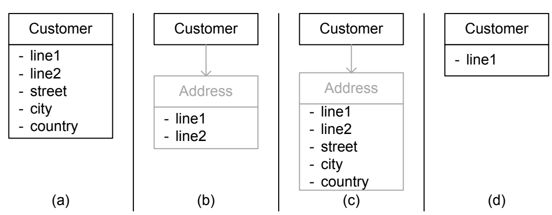


Fig. 3. Attribute synchronization

This is the case of class `Customer`′ in Fig. 3 (b). There are attributes `line1`′ and `line2`′ with interpretations `line1` and `line2`, respectively, and with the interpreted class context `Customer`′. Therefore, the propagation creates the attributes `street`′, `city`′, and `country`′ with interpretations `street`, `city`, and `country` with the interpreted class context `Customer`′, respectively as shown in Fig. 3 (c). On the other hand, there is only a single attribute with an interpretation `line1` in Fig. 3 (d) and no attribute with interpretation `line2`. Therefore, the synchronization is not propagated in this case. Due to lack of space we do not show the propagation of synchronization of associations. However, the process is very similar to the synchronization of attributes.

6 Evaluation

The conceptual modeling language for XML proposed in our previous works was implemented in a case tool *eXolutio*[1] including the system of atomic and composite

[1] `http://www.eXolutio.com`

operations described in this paper. We used the implementation to evaluate our approach on XML vocabulary standardized by Czech public authorities. It is used in a public procurement system and its PIM schema is simple. It has 4 classes interconnected by 9 associations. The vocabulary comprises 17 XML schemas each modeled by a separate PSM schema. Figure 4(a) shows the number of atomic operations per-

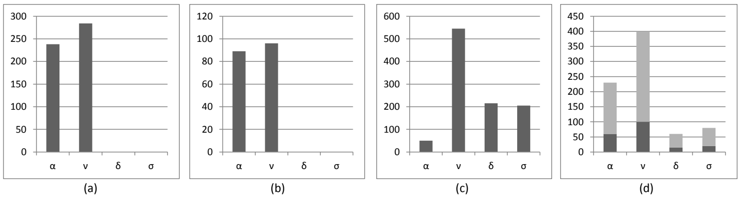


Fig. 4. Statistics

formed to create the PIM and PSM schemas. Only creation and sedentary operations were necessary. First, there was a requirement to model 6 new XML schemas as 6 additional PSM schemas. The number of performed atomic operations is depicted in Figure 4(b). Only creation and sedentary operations were performed. Second, there was a requirement to make the existing XML schemas more readable for developers. This required renaming some existing XML elements and attributes in the XML schemas and their reconnecting. Also merges of various components into single ones were done. In both cases, only local changes in PSM schemas were done and the PIM schema was not affected. The number of performed atomic operations is depicted in Figure 4(c). All kinds of operations were necessary in this step. Third, various changes to the PIM schema needed to be made. These changes resulted from a new legislation which is currently implemented in Czech republic. In this case, the changes were correctly propagated by our introduced mechanism to the PSM schemas. The number of performed atomic operations is depicted in Figure 4(d). The study showed us that all introduced atomic operations are necessary. Also, they are sufficient for all kinds of changes performed. The light gray columns in Figure 4(d) show the number of atomic operations automatically performed by our propagation mechanism. Without the mechanism, they would have to be performed manually by the designer. However, the amount of manual work saved is much higher. Even in the previous steps, it saves a significant amount of work. When the designer designs a new PSM schema on the base of the PIM schema, our technique ensures that he works consistently with the PIM schema and, therefore, with other PSM schemas. The designer does not need to check this consistency manually which saves him a great deal of work and prevents from design errors.

7 Conclusions

In this paper we have identified several problems of design and evolution of a family of XML schemas and showed how to solve them using the strategy of MDA. We showed that a common conceptual schema in a PIM may be designed and the XML schemas may be then derived in a form of visual schemas of PSM. We defined the sets of atomic operations for the two levels and demonstrated their propagation. We show on preliminary experiments that our strategy apparently enables to save both manual effort and errors of evolution management. In our near future work, we plan to further develop our tool and evaluate it in various real-life projects we are currently planning.

References

1. HL7, http://www.hl7.org/
2. OpenTravel.org, http://www.opentravel.org/
3. Al-Jadir, L., El-Moukaddem, F.: Once Upon a Time a DTD Evolved into Another DTD.. In: Masood, A., Léonard, M., Pigneur, Y., Patel, S. (eds.) OOIS 2003. LNCS, vol. 2817, pp. 3–17. Springer, Heidelberg (2003)
4. Domínguez, E., Lloret, J., Rubio, Á.L., Zapata, M.A.: Evolving XML Schemas and Documents Using UML Class Diagrams. In: Andersen, K.V., Debenham, J., Wagner, R. (eds.) DEXA 2005. LNCS, vol. 3588, pp. 343–352. Springer, Heidelberg (2005)
5. Klettke, M.: Conceptual XML Schema Evolution – The CoDEX Approach for Design and Redesign. In: BTW 2007, Aachen, Germany, pp. 53–63 (March 2007), http://dbs.cs.uni-duesseldorf.de/BTW2007/Klettke.pdf
6. Malý, J., Mlýnková, I., Nečaský, M.: XML Data Transformations as Schema Evolves. In: Eder, J., Bielikova, M., Tjoa, A.M. (eds.) ADBIS 2011. LNCS, vol. 6909, pp. 375–388. Springer, Heidelberg (2011)
7. Mesiti, M., Celle, R., Sorrenti, M.A., Guerrini, G.: X-Evolution: A System for XML Schema Evolution and Document Adaptation. 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. 1143–1146. Springer, Heidelberg (2006)
8. Miller, J., Mukerji, J.: MDA Guide Version 1.0.1. Object Management Group (2003)
9. Nečaský, M.: Conceptual Modeling for XML, Dissertations in Database and Information Systems, vol. 99. IOS Press, Amsterdam (2009)
10. Nečaský, M., Mlýnková, I.: When Conceptual Model Meets Grammar: A Formal Approach to Semi-structured Data Modeling. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 279–293. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-17616-6_26
11. Nečaský, M.: Conceptual modeling for XML: A survey. In: Snasel, V., Richta, K., Pokorny, J. (eds.) DATESO. CEUR Workshop Proceedings, vol. 176, CEUR-WS.org (2006), http://www.ceur-ws.org/Vol-176/paper7.pdf
12. Nečaský, M., Malý, J., Klímek, J., Mlýnková, I.: Evolution and Change Management of XML Applications. Tech. Rep. XML and Web Engineering Research Group, Charles University in Prague (May 2011), http://www.eXolutio.com/TR/evolution.pdf
13. Passi, K., Morgan, D., Madria, S.: Maintaining integrated XML Schema. In: IDEAS 2009, pp. 267–274. ACM, New York (2009)
14. Su, H., Kramer, D., Chen, L., Claypool, K., Rundensteiner, E.A.: XEM: Managing the Evolution of XML Documents. In: RIDE 2001, pp. 103–110. IEEE, Washington, DC (2001)
15. Tan, M., Goh, A.: Keeping Pace with Evolving XML-Based Specifications. In: Lindner, W., Fischer, F., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 280–288. Springer, Heidelberg (2004)
16. Yu, C., Popa, L.: Semantic Adaptation of Schema Mappings when Schemas Evolve. In: VLDB 2005, pp. 1006–1017. VLDB Endowment (2005)

Towards a Characterization of Egocentric Networks in Online Social Networks*

Valerio Arnaboldi, Andrea Passarella, Maurizio Tesconi, and Davide Gazzè

IIT-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy
firstname.lastname@iit.cnr.it

Abstract. Online Social Networks (OSNs) are more and more establishing as one of the key means to create and enforce social relationships between individuals. While substantial results have been obtained in the anthropology literature describing the properties of human social networks (built "outside" the OSN world), a clear understanding of the properties of social networks built using OSNs is still to be achieved. In this paper we provide a contribution towards this goal, by starting characterizing ego networks formed inside Facebook through the analysis of data obtained from a measurement campaign. Ego networks capture all the social relationships (links) between an ego and all the other people with whom the ego has a social tie. Ego networks are one of the key social structure that have been studied in the anthropology literature, and is thus a reference objective for our work. In this paper we analyze a number of quantitative variables that can be collected on Facebook, which can be used to describe the properties of the social links in ego networks. We also analyze the correlation between these variables and the strength of the social ties, as explicitly ranked by the monitored Facebook users. Our results show that there is an interesting similarity between the properties observed by anthropologists in human social networks, and those of Facebook social networks. Moreover, we found a noticeable correlation between most of the measured variables and the tie strengths, suggesting the possibility of automatically inferring the latter from measurable Facebook variables.

Keywords: Online Social Networks, Measurements, Ego networks.

1 Introduction

Social network analysis has become one of the most important multidisciplinary methodology aimed at studying people relationships and information flow within groups of individuals. Social structures are represented as networks of nodes and edges, modelling the set of individuals in a certain context and all the social ties existing between them.

* This work was partially funded by the European Commission under the FIRE SCAMPI (FP7-258414), FET-AWARE RECOGNITION (FP7-257756) and CAPER (FP7-261712) Projects.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 524–533, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Ego networks are a particular kind of social networks representation in which the only relationships considered are those between one person (called ego) and all the individuals directly connected to her (alters). These networks are one of the reference representations of human personal social networks and are largely studied in the anthropology literature [7,6,11,14]. Given two individuals, their tie strength is a numerical representation of the importance of their relationship and is typically used as the weight of the link between them in the social network. The challenging issue in the representation of social networks in general - and of ego networks in particular - is how to find a suitable way to determine the ties strength, given that it should embody all the possible factors underlying social relationships, which can be difficult to be identified.

In [10], the author has given the first informal definition of tie strength, conjecturing the presence of more than one dimension beneath social ties. The author identifies these dimensions as the amount of time, the emotional intensity, the intimacy and the reciprocal services which characterize the relationships. Successive work have added more dimensions to the first definition, achieving a more accurate tie strength characterization. In [2] structural variables such as the number of mutual friends or the dimension of the network are considered as important factors controlling the behaviour of a social relationship. In [11] authors found the frequency of contact to be a good predictor for the social tie strength. Other possible factors controlling for the tie strength are gender, with a direct impact on non face-contact, age and kinship [6].

The diffusion of online social networks (henceforth OSNs), like Facebook, Twitter, LinkedIn and many others, is fostering the availability of a lot of data regarding social interaction between people which was impossible to obtain until few years ago. Several early conjectures made by sociologists on social networks have been confirmed by the analysis of empirical data obtained from the Internet [4,12,13]. While ego networks have been well analysed in the anthropology literature, the characterization of ego networks formed in OSNs has received little attention so far [9,8]. In this paper, we start filling this gap by analyzing the results of a measurement study on Facebook users. The overall goal of this work is starting investigating whether the structure of ego networks formed in OSNs (considering Facebook as representative of them) is similar or not to the structure of ego networks observed in real human networks by anthropologists [7,6,11,14]. To this end, we measured two classes of variables, i.e. *socio-demographic* variables, such as the size of the ego network of the users and various factors possibly impacting on it (e.g., age, gender, ...), and *relational* variables, which relate to the strength of the social relationships between each ego and her alters. In Section 2 we describe in detail our experimental methodology.

In the paper we analyze the distributions of the socio-demographic variables across the users, and of the relational variables across the social relationships of each user. While measurement studies of Facebook users are available in the literature [8,9], to the best of our knowledge this is the first paper in which the distributions of the quantitative variables related to the ego networks and tie strength dimensions are analyzed. Moreover, the paper also presents an initial correlation analysis, showing how well each of the measured relational variables correlates with the real tie strength of social relationships. As described in Section 2, real tie strengths have been collected by asking users to explicitly rank the importance of their friends "in their Facebook world".

The key findings of the paper can be summarized as follows (the data and a detailed discussion on their properties are presented in Sections 3 and 4). First of all, we find that the average size of ego networks is in the range of human network sizes found in the anthropology literature [7,11,14,16]. Moreover, the analysis of Facebook variables distributions clearly shows that, from the perspective of an ego, there exist a relative small fraction of "very important" alters, with whom the ego interacts far more than with the rest. As similar features have also been found in human social networks, these results suggest that the structure of OSN ego networks might be quite similar to that of human social networks. Finally, the correlation analysis confirms that the measured variables are sensibly correlated with the real tie strength perceived by the users, and allows us to start understanding which variables, among the measured ones, can better predict tie strengths. This is an initial step towards more refined models, able to automatically estimate tie strengths, only based on the patterns of interactions between users.

2 Data Acquisition

The detailed list of Facebook variables we have identified as best descriptors for ego networks characteristics is listed in table 1. We decided to consider only quantitative variables and to discard all user-filled quantities, which would be prone to the typical problems of semantic analysis.

Relational variables are expected to be related to the tie strength of the corresponding social relationship. Because the tie strength is not directly measurable from Facebook data, this relation is unknown. Thus we decided to collect values of tie strength perceived by the users, asking them to rate their friendships answering the question: *"How do you rate, with a value between 0 and 100, the social relationship between you and this person in Facebook?"*. We collected the answers through an electronic survey. Using a generic question allowed us to capture all the possible definitions of tie strength. With this data we have been able to analyze the correlation between Facebook variables and the social tie strength.

As in [7], we distinguish between active and non-active friends. Active friends are those alters for whom ego spends a non negligible effort to maintain their relative relationships alive. To identify active friends we decided to take into account the set of people who have received a value of tie strength larger than zero. We only considered active friends in our data analysis, leaving the relation between active and non-active networks for future work.

To obtain the data sample for our analysis we performed data acquisition campaign involving 30 people (18 males and 12 females) to whom we asked to use a Facebook application we have built, named Facebook Analyser (FBA).

Table 1. Facebook variables chosen as ego network descriptors

Variable type	**Variable name**
Socio-demographic	age
	gender
	relationship status
	number of friends
	number of active friends
	total number of status updates made
Relational	number of exchanged posts
	number of exchanged private messages
	number of exchanged likes
	number of exchanged comments
	number of mutual friends
	number of tags on the same pictures
	number of days since first communication
	number of days since last communication
	number of tags on the same objects
	number of events attended together
	number of groups in common
	number of likes to the same pages

FBA is able to collect both Facebook variables listed in table 1 and the values of tie strength perceived by the users towards all their friends, with an embedded electronic survey. On average, FBA takes about half an hour for Facebook data acquisition and few minutes for tie strength evaluation. The total duration of data acquisition campaign was three weeks. We have collected a total of 7665 relationships, from which we have extracted 3245 active friendships. While the number of social relationships we have sampled is significant, the number of users involved in the experiment is not sufficient to draw definite conclusions. However, as discussed in section 3, this sample is already sufficient to provide interesting indications on the properties of OSN ego networks, and their similarity with ego networks observed in human social networks.

To study correlations between involved variables, we use the sample correlation coefficient r, which, given two random variables X and Y, is defined as:

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{(n-1)S_x S_y} , \tag{1}$$

where $\overline{x}$ and $\overline{y}$ are the sample mean, and S_x and S_y are the sample standard deviations, of X and Y, respectively. This is an estimator of the Pearson product-moment correlation coefficient (also known as Pearson's r), defined as:

$$\rho_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y} , \tag{2}$$

where $COV(X, Y)$ is the covariance and σ_X and σ_Y are the standard deviations. The p values presented with the correlation values regard the correlation significance. Small p-values indicate that X and Y are correlated (see [15] for a precise definition of p).

3 Data Analysis

In this section we present the data obtained from our measurements. A discussion of the main properties that emerge from them is postponed to Section 4.

3.1 Socio-demographic Variables

Our experiment involved people randomly chosen within our research area. All the participants were researchers, Ph.D students or master students from 24 to 48 years old ($M = 33.17, SD = 6.78$). One of the aspects we are most interested in is the mean ego-network size. Facebook provides a tool to measure the average number of friends of users, which turns out being around 130. However, this number also considers non-active relationships and unused accounts, thus it can not be used as a reliable measure for our purposes.

The total number of Facebook friends - both active and non-active - of the participants involved in our experiment varies between 59 and 1099 ($M = 255.5$, $SD = 197.59$). However, the active network size ranges between 29 and 368 ($M = 108.17$, $SD = 85.55$). The distribution of this variable, which is depicted in figure 1, is similar to that found by anthropologists [14], [11], studying the properties of human social networks. The mean active network size is also of the same order of that found in human social networks (e.g., 124 in [11]). This number also shows significant correlation with the mean density of the ego networks. ($r = 0.48$, $p < 0.01$). Age does not show a significant correlation with network size. This result is in accordance with [6].

These results provide a first indication of a similarity between the ego networks found in human social networks and those formed by OSNs users.

3.2 Relational Variables

To analyse relational variables we split them into more subclasses, to better study the relation between different kinds of interaction and the social tie strength. The classes we have identified are: i) text-based communication variables, ii) like-based communication, iii) homophily variables, iv) time variables and v) structural variables. While most of these classes are self-explanatory, the homophily concept needs a better definition. Homophily measures the social similarity between two individuals. It is the tendency of people with similar social characteristics to have stronger social ties compared to dissimilar individuals.

Text-based communication variables capture all kinds of textual interactions between individuals which can be exploited on Facebook. These variables identify communication styles similar to other traditional methods, such as mail, text

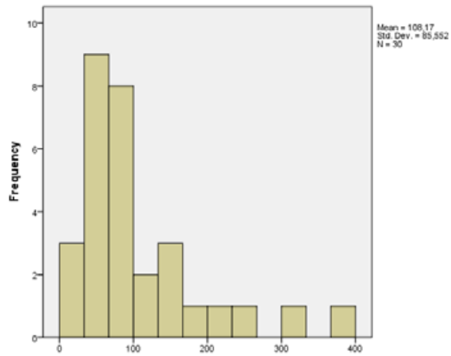
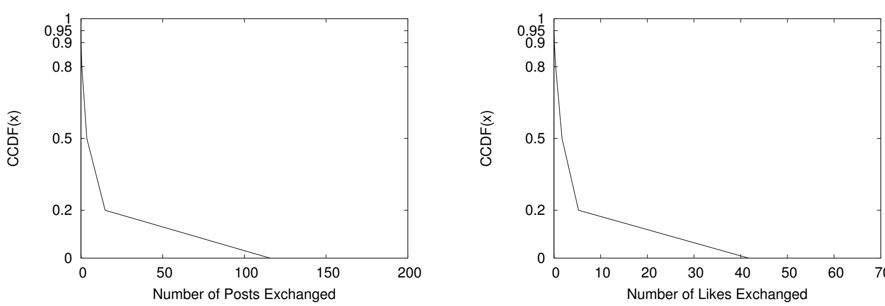


Fig. 1. Active Network Size

messages etc. largely studied by other authors [11] [3]. In these works the evidence of a relation between textual interaction and tie strength has been found. In this class we place variables representing the number of posts, comments and private messages exchanged within Facebook. Figure 2a depicts the CCDF of the number of posts exchanged between ego and alters in our sample. Specifically, we have obtained the indicated percentiles for each user, and averaged them over all users. We don't present the CCDF of the other text-based variables because they show a behaviour similar to posts, which is the most representative variable of the class. Text-based interaction variables show a medium correlation with the perceived tie strength ($r = 0.2$, $p < 0.01$ for comments, $r = 0.2$, $p < 0.01$ for private messages and $r = 0.39$, $p < 0.01$ for posts).



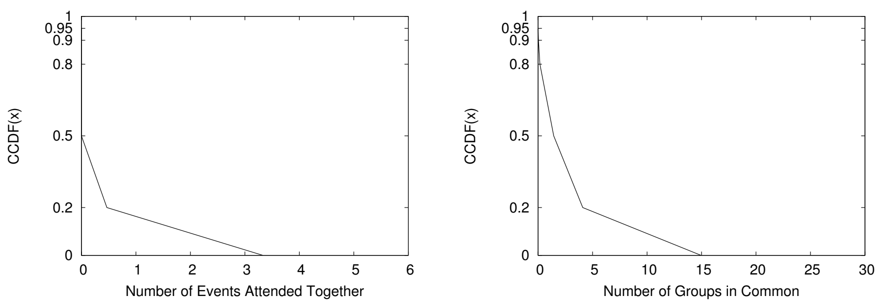
(a) CCDF for the number of posts exchanged between individuals in Facebook

(b) CCDF of the number of likes exchanged between individuals in Facebook

Fig. 2. CCDF of posts and likes

The second class of variables encompasses a novel type of communication, largely used in Facebook, called "like". Likes are special marks left by users on Facebook objects to express a positive feedback. These objects can be posts, comments, pictures, videos and other. The number of likes received from or made to a certain person could be a good predictor for emotional support, intimacy and frequency of contact of the relationship. Figure 2(b) depicts the CCDF regarding the number of likes exchanged between egos and their friends. As for textual interaction, likes show a long tailed shape. Also like-based communication variables show medium correlation with the tie strength ($r = 0.35$, $p < 0.01$). This result is similar to that previously shown for private messages exchange and displays the high importance of like-based communication inside Facebook.

In the class of homophily variables we have placed the number of groups to which a pair of Facebook friends have a subscription in common, the number of events attained together, the number of likes made on the same objects and the number of pictures in which users appear together. To discover this last quantity we rely on a particular kind of action, typical of the OSNs, called tagging. Tagging is a way to mark a picture with the name of one or more people who appear in it. The presence of two people in the same picture or attending the same event can be seen as good predictor not only for homophily, but also for intimacy and emotional closeness. In Figure 3 we show the CCDF regarding the number of common events and groups (CCDFs of likes on the same objects and tags on the same pictures are qualitatively similar). Homophily variables are more or less equally correlated to tie strength ($r = 0.24$, $p < 0.01$ for events, $r = 0.25$, $p < 0.01$ both for groups and likes).

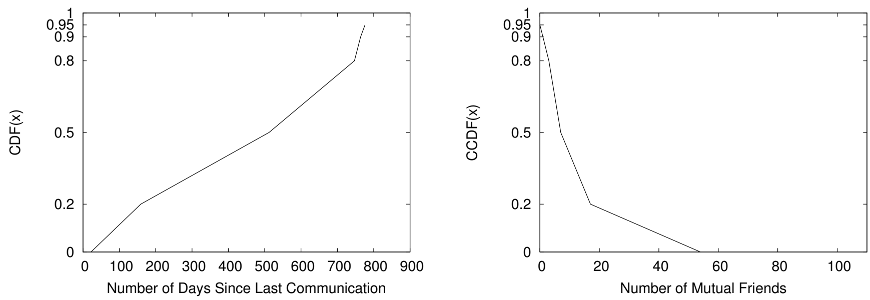


(a) CCDF of the number of events two Facebook friends have attended together

(b) CCDF of the number of groups two Facebook friends have in common

Fig. 3. CCDF of common events and groups

Time variables indicate the frequency of contact and the duration of a relationship. We have considered the time elapsed since the last communication and the time since the first communication, considering all the text-based and like-based interaction variables. Time variables facilitate the distinction between long and short-timed relationships and can discover fading out of social ties - for example if time to first contact and time to last contact are both high presumably the relationship has decayed. Figure 4(a) depicts the CDF of the number of days elapsed since the last communication made between two Facebook users. Time variables are correlated, as expected, with tie strength ($r = 0.23$, $p < 0.01$ for time since first communication and $r = -0.3$, $p < 0.01$ for time since last communication). The negative correlation between time to last contact and tie strength is due to the fact that if the time period elapsed since the last interaction between two people is short, the probability of having a strong tie between them is higher. Time to last contact has been used as the predictor of emotional closeness in most of the work present in the anthropological literature [7,11,14].



(a) CDF of the number of days elapsed since the last communication traced between two Facebook friends

(b) CCDF of the number of mutual friends between pairs of Facebook friends

Fig. 4. CDF of the number of days since last communication, and CCDF of mutual friends

The last class of variables contains all the structural dimensions of ego-network ties obtainable from Facebook data. In this class we have identified the number of mutual friends and the age difference between peers as best predictors for tie strength. Age difference between ego and her friends does not show a significant correlation with tie strength. The number of mutual friends is weakly correlated with tie strength ($r = 0.23$, $p < 0.01$). In Figure 4(b) the CCDF of the number of mutual friends is depicted.

4 Discussion

The CCDF of all the relational variables, shown in Figure 2, 3, and 4 present a long-tailed shape. This result suggests the presence of a small set of alters tightly connected to ego and a larger group of people loosely tied to her. Note that this also results for the CDF of the time since last contact, as there is a clear distinction between a set of frequently contacted alters and the rest. These results are similar to the findings in [11] related to human social networks.

The findings in [5] made us intuitively hypothesize the presence of a positive correlation between contact frequency and tie strength. Our results confirm this fact, both in terms of time to last contact and total number of contacts. Comparing the values of correlation seen so far we can say that homophily variables show an unexpected minor influence on tie strength compared to text and like-based communication variables or time to last contact. Nevertheless homophily could play an important role in the formation of ties rather than in the determination of their weight.

We have found a significant and positive correlation between active network size and network density (i.e. the mean tie strength within the network). This result is in contrast with the findings in the anthropological literature, which have confirmed the tendency of larger network to be less dense [14]. This difference could be explained taking into account that we are studying virtual ego networks, which are only a subset of individuals' complete social networks and users with few friends might not use Facebook to maintain their strongest relationships. Moreover, users with more Facebook friends might spend more time using Facebook than people with few friends, augmenting the weight of the relationships with them. This intuition should be validated with dedicated studies.

As in [6], we have not found a significant correlation between ego age and the size of her network, yet we haven't discovered significant correlation between age and network density (as observed in [6]). This could be due to the low age range of our sample. Authors of [6] have considered a sample composed of participants ranging from 18 to 65 years old ($M = 38.5$, $SD = 13.3$) and they have found significant differences between people placed at the extremes of the sample. Our sample is more homogeneous and does not include those extreme values.

5 Conclusions

In this paper we have presented an initial yet detailed characterization of virtual social ego networks through the analysis of a set of observable Facebook variables. We made a selection over all possible Facebook obtainable information to have a set of variables which at least can contain all the social tie strength dimensions already identified in the sociological and anthropology literature [10,1,2], and confirmed by recent empirical studies [9,8].

We have gathered the values of tie strength perceived by the participants of our experiment towards all their Facebook friends thanks to an electronic survey embedded into our Facebook application. Hence we have analyzed the distributions of the observed variables, and the relation between them and tie strength through correlation analysis.

The results we have found indicate that Facebook active ego networks obtained from our sample data have distribution and mean size similar to those found in other work [14,11,16] in the anthropology literature. The distributions of the relational variables we have taken into account have long tailed shapes. This evidence confirms the presence of an high number of weak ties and a lower number of strong ties, as found in [7,14,11].

The correlation analysis revealed the relation between the Facebook observable quantities we have considered and the sampled tie strength. Therefore, the relational variables we have identified are good proxies for tie strength. Like-based communication has shown correlation value comparable with all the other types of communication available on Facebook. This is an evidence of the importance of this style of communication, which is becoming more and more popular in OSNs.

In this work we have not considered the possibility to predict tie strength using the observable quantities we have identified, as done in [9]. As subject of future work, we will target this issue to automatically obtain ego networks representation without asking the users to rate all their friendships. Moreover, we will investigate if specific structures can be identified in OSN ego networks, as it is the case in human social networks.

References

1. Barry, W., Wortley, S.: Different Strokes From Different Folks. American Journal of Sociology 96(3), 558–588 (1990)
2. Burt, R.S.: Structural Holes versus Network Closure as Social Capital, pp. 31–56, May 2000 (2001)
3. Culotta, A., Bekkerman, R., Mccallum, A.: Extracting social networks and contact information from email and the Web. In: Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. No. d (2004)
4. Dodds, P.S., Muhamad, R., Watts, D.J.: An experimental study of search in global social networks. Science 301(5634), 827–829 (2003)
5. Dunbar, R.I.M.: The social brain hypothesis and its implications for social evolution. Annals of Human Biology 36(5), 562–572 (1998)
6. Dunbar, R.I.M., Roberts, S.: Communication in Social Networks: Effects of Kinship, Network Size and Emotional Closeness. Personal Relationships (2010)
7. Dunbar, R.I.M., Spoors, M.: Social Networks, Support Cliques and Kinship. Human Nature 6(3), 273–290 (1995)
8. Ellison, N.B., Steinfield, C., Lampe, C.: The Benefits of Facebook Friends: Social Capital and College Students Use of Online Social Network Sites. Journal of Computer-Mediated Communication 12(4), 1143–1168 (2007)
9. Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: International Conference on Human Factors in Computing Systems. ACM Press, New York (2009)
10. Granovetter, M.S.: The Strength of Weak Ties. The American Journal of Sociology 78(6), 1360–1380 (1973)
11. Hill, R.A., Dunbar, R.I.M.: Social network size in humans. Human Nature 14(1), 53–72 (2003)
12. Leskovec, J., Horvitz, E.: Planetary-Scale Views on an Instant-Messaging Network. Tech. rep. (2007)
13. Onnela, J.P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., Kertész, J., Barabási, A.L.: Structure and tie strengths in mobile communication networks. National Academy of Sciences of the United States of America 104, 7332–7336 (2007)
14. Roberts, S.G., Dunbar, R.I., Pollet, T.V., Kuppens, T.: Exploring variation in active network size: Constraints and ego characteristics. Social Networks 31(2), 138–146 (2009)
15. Sirkin, R.M.: Statistics for the Social Sciences (2006)
16. Zhou, W.X., Sornette, D., Hill, R.a., Dunbar, R.I.M.: Discrete hierarchical organization of social group sizes. Biological Sciences 272, 439–444 (2005)

Privacy Preserving Gate Counting with Collaborative Bluetooth Scanners

Nelson Gonçalves[1], Rui José[2], and Carlos Baquero[3]

[1] Universidade do Minho, Portugal
[2] Centro Algoritmi, Universidade do Minho, Portugal
rui@dsi.uminho.pt
http://www.dsi.uminho.pt/~rui/
[3] DI/CCTC, Universidade do Minho, Portugal
cbm@di.uminho.pt
http://gsd.di.uminho.pt/cbm/

Abstract. Due to its pervasiveness and communication capabilities, Bluetooth can be used as an infrastructure for several situated interaction and massive sensing scenarios. This paper shows how Bluetooth scanning can be used in gate counting scenarios, where the main goal is to provide an accurate count for the number of unique devices sighted. To this end, we present an analysis of several stochastic counting techniques that not only provide an accurate count for the number of unique devices, but offer privacy guarantees as well.

Keywords: Privacy, Gate Counter, Bloom Filters, Hash Sketches.

1 Introduction

As Bluetooth becomes more and more pervasive, there is a growing potential to leverage on the possibilities offered by Bluetooth scanning as a flexible infrastructure for situated interaction and a general purpose platform for massive sensing and actuation in urban spaces. Bluetooth sensing is based on a discovery process through which a device can inquire about the presence of other nearby devices. If those devices are in *discoverable* mode, they will respond with their address, and possibly additional information, such as the device name, the device type (e.g. cellphone or computer) and available services. A Bluetooth scanner is a device that periodically scans nearby devices, registering and timestamping the observations and making them available to other applications and systems. Multiple Bluetooth scanners spread all over a city could thus serve as collection points for Bluetooth sightings, providing a major tool for observing, recording, modelling and analysing the city, physically, digitally and socially [8].

A large scale infrastructure of that nature is not likely to emerge as a single-domain initiative. There would not be a killer application that could by itself justify such huge investment. However, there is already a large number of Bluetooth scanners in urban environments. They are owned by many entities and

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 534–543, 2011.
© Springer-Verlag Berlin Heidelberg 2011

they serve very diverse purposes, such as proximity marketing, device localization or OBEX-based interaction. These same scanners could be used, without any additional cost, as nodes in a large scale collaborative sensing infrastructure. Each node would still scan for its own purposes, but it would also share part of the generated data with a central service that would then be able to produce aggregate information of mutual interest. This collaborative path could enable large scale Bluetooth sensing to quickly enter mainstream.

A major obstacle, however, is how to enable this type of large scale sensing without creating an overwhelming privacy threat. A Bluetooth scanner registers the Bluetooth addresses, 48-bit MAC, of the devices that have been sighted. Some devices are able to switch between multiple Bluetooth addresses, but for most cases this will be a reliable and permanent unique identifier for sighted devices and by extension to the respective owners. While a single scan of proximate devices is not in itself much of a problem, a systematic registration of Bluetooth sightings, especially when done at multiple locations, would have the potential to become a large scale tracking system. Relatively simple processes could be put in place to detect the presence, movements and patterns of individuals as well as co-location patterns between people. For privacy, a precautionary measure should avoid permanent storage and dissemination of Bluetooth addresses.

1.1 Bluetooth-Based Collaborative Gate Counting

In this study, we address one of the most common forms of urban sensing: counting unique visitors across multiple gate counters to measure the flow of people across the urban setting. In this scenario, a potentially very large set of heterogeneous and autonomous nodes support the counting process by acting as Bluetooth-based gate counters.

Conceptually, a gate is a virtual line across a street, and gate counting is the process of counting the number of people crossing that line. Each Bluetooth node is modeled as a gate counter that counts the number of unique Bluetooth addresses observed during a certain period. The use of Bluetooth as an enabling technology for gate counting has been extensively explored in [8] to establish the flows of people at sampled locations within a city over the course of a day. A Bluetooth-based gate counter does not really count all the persons passing-by, but only those who are carrying discoverable Bluetooth devices. Still, this is enough to make a reasonable correlation, using baseline data, to estimate the overall traffic.

A Bluetooth-based gate counter needs to recognize subsequent sightings of the same entity. Repeated sightings of the same device can be very common, not only because people can be passing-by multiple times, but also because of persistent devices. Instead of scanning through a line, discovery is actually performed in an area and thus any device in that area, possibly in nearby buildings, would be repeatedly discovered. Results from empirical studies with known static and transient devices suggest that a transient device typically appears for up to 90 seconds while it crosses a gate [8]. A proper gate counting process would thus needs to account for this and filter persistent devices.

A gate counter should also be able to answer questions like "how many different people were seen in a given gate in the last 24 hours ?" or "what was the number of visitors of an amusement park during visitors peak hours?". To comply with this requirement, gate counters should be able to distinguish its readings over time.

The main challenge, however, is how to enable collaborative counting between multiple independent gate counters, while providing appropriate privacy guarantees as well as low communication costs. To be able to count unique entities, we need to identify multiple counts of the same entity at different nodes, to make sure that the same device sighted at two different gates will be counted only once. Imagine, for example, a city festival with multiple gate counters operating at various locations to count the number of visitors to the festival. The simple sum of individual gate counts would clearly overestimate the number of people since many of them would be spotted at multiple gates. We thus need some technique that works across multiple gate counters and is able to provide an aggregate count of the unique device addresses that have been spotted in the entire set of gate counters. Comparing the plain addresses observed at different gates would immediately solve this problem, but is not appropriate for privacy preservation.

1.2 Objectives

In this paper, we explore the use of stochastic summarizing techniques as a privacy preserving approach to enable Bluetooth-based gate counting of unique entities across multiple nodes. The objective is to assess to what extent these techniques are able to address the specific requirements of this distributed gate counting model and inform the design of large scale Bluetooth sensing systems.

We have identified the sensing requirements for this scenario and established a number of key criteria for assessing the various alternatives. We have then conducted an experimental study in which we compared how multiple types of stochastic summarizing techniques would behave across multiple variants of our gate counting scenario. The results provide a strong foundation for the development of these large scale Bluetooth sensing infrastructures, identifying major trade-offs and the implications of key factors such as cardinality and support for merge operations.

2 Probabilistic Counters for Privacy-Enhanced Gate Counting

A common strategy in privacy-enhancing techniques is to reduce the data collected to what is absolutely needed for a particular purpose. In this gate counting scenario, we need to count devices and the ability to check if a particular device has already been counted before. Moreover, we need to be able to do this across a set of autonomous nodes. This means that we are not interested in information about the number or duration of sessions that each device generates at each gate.

Probabilistic counters provide an interesting solution to this problem. They are flexible enough for estimating the overall number of unique sightings with some controllable accuracy, without ever keeping the plain Bluetooth addresses.

In this section, we make a brief description of the algorithms that were implemented and tested for the Gate Counter scenario.

2.1 Bloom Filters

Bloom Filters were created in 1970 [2] by Burton Howard Bloom. They are a simple and memory efficient probabilistic data structure for set representation where membership queries are allowed. A Bloom Filter consists in a bit array (initially set to 0) and k consistent hash functions. Those hash functions are used to map an element into several array positions. The bits at those positions are then either set to 1 (in order to add the element to the Filter), or checked to see if all are set to 1 (in case they are, the element is considered part of the set). In short, the main properties of Bloom Filters are:

- Small memory footprint in comparison with the memory needed to represent the actual set,
- The add and check membership operations have $O(k)$ complexity, where k is the number of hash functions, therefore independent from the number of elements represented in the set.
- False positives are possible, but their occurrence can be controlled,
- No false negatives.

In this particular Gate Counter Scenario we used both the standard version of Bloom Filters[2] and Scalable Bloom Filters [1]. Scalable Bloom Filters are a variant of Bloom Filters that can dynamically adapt to the number of elements stored while respecting a maximum false positive probability, which is chosen at the beginning. Even though it's not their main goal, Bloom Filters (and Scalable Bloom Filters) can also be used to estimate the cardinality of multisets. This is accomplished using the ratio of bits set to 1 in the bit array. The size m of the bit array supporting the bloom filter is linear, $O(N)$, with the number N of elements to count.

2.2 Hash Sketches

Hash Sketches are a simple probabilistic data structure with which we can obtain the cardinality of sets. Much like Bloom Filters there are several variants of this algorithm. Despite being different, all of these variants have at least one bit array and use some kind of hash function to map elements to positions in the aforementioned array(s).

Hash Sketches have a small memory footprint, the ability to estimate the cardinality in a single pass over the set, as well as $O(1)$ complexity to add a new element and $O(m)$ complexity to estimate the cardinality (where m is the size of the bit array).

In the Gate Counter Scenario, we tested several versions of sketches: LogLog Sketches [3], HyperLogLog Sketches [7], Linear Counting Sketches [9], Robust In Network Aggregation Linear Counting Sketches (RIA-LC) [5,4] and Robust In Network Aggregation Dynamic Counting Sketches (RIA-DC) [4].

LogLog Sketches are similar to the Probabilistic Counting algorithm presented in [6] since both use several small bit arrays (called buckets) instead of a single bit array. The main difference is that LogLog Sketches are much less memory consuming at the expense of some accuracy. Their name derives from the fact that each small bit array has size close to $log(log(N))$, being N the number of distinct elements. The estimate of the cardinality is obtained using the average of the several small bit arrays.

HyperLogLog Sketches are an improvement over LogLogSketches. Using the same number of bits as LogLog Sketches, HyperLogLog Sketches are able to provide more accurate results. According to the authors in [7], this improvement is accomplished by using *harmonic means* instead of *geometric means* in the evaluation function.

Linear Counting Sketches use only a single bit array. Their name comes from the fact that they have $O(N)$ size, meaning their size grows linearly with the number of distinct elements N. When compared to LogLog Sketches, in Linear Counting Sketches the size is a drawback, but they work better for small cardinality sets.

Both RIA-DC and RIA-LC sketches are based on the Linear Counting Sketches. While RIA-LC can be seen as a slightly improved/simplified version of Linear Counting Sketches, RIA-DC Sketches have the unique ability to merge sketches of different sizes. However, this ability comes with a price. RIA-DC Sketches assume that there is no overlap of elements belonging to different sketches, meaning that if we merge two different RIA-DC sketches with elements in common, those elements will be counted twice in the final aggregate.

3 Comparative Analysis

The presented techniques, should now be evaluated against the specific requirements of gate counting scenarios. Considering the requirements identified in subsection 1.1 we will now analyze the proposed techniques against the following criteria: **accuracy**, **size** and **aggregation**.

3.1 Criteria

Accuracy. With this criterion we want to evaluate the accuracy of the techniques, their ability to count multiple sightings of the same device only once, and the quality of their estimators. Setting the maximum standard error to 5%, $\sigma = 0.05$, we measured for all techniques the relative error (root min squared error) for a range of cardinalities, whose average of 100 runs is shown in Fig 2.

Size. The size of the techniques is an important factor. The less space the technique requires, the lower will be both the costs of communication between BT scanners and their memory requirements.

Regarding this criterion, we have 2 fundamentally different types of techniques: dynamic techniques which consist of Scalable Bloom Filters and static techniques comprising the rest. Static size techniques are techniques whose size is set at the time of creation and cannot be changed afterwards. This means that we must know the maximum number of unique devices to count before hand, or at least, we must be able to assume an upper bound for that number. Dynamic techniques on the other hand don't have this drawback because they can adjust to withstand arbitrarily large cardinalities. In practical terms this means that for static size techniques, once we create an instance with a certain capacity it is not possible to change that capacity afterwards, while for dynamic techniques there is no such constraint.

To further help us in our analysis, we can look at Figures 3(a) and 3(b) which respectively depict the number of bits required by each unique element and the size spent by each technique for a range of cardinalities.

Aggregation. The ability to merge counts is crucial for scenarios with multiple gates. It is a key ingredient for obtaining the *aggregate number* of individuals in a set of gates. The merge operation consists in either a bitwise OR operation (Bloom Filters, Linear Counting, RIA-DC and RIA-LC sketches) or in a max operation (HyperLogLog and LogLog sketches) of structures that make each gate's counter. In order to merge several gate counters, there are 2 conditions that must be met: all counters must be instances of the same technique and every instance must have the same parameters and capacity (equally sized bit arrays). Meeting these conditions ensures that the same unique device will mark the same positions in the several gate counters it crosses. Therefore after merging (bitwise OR or max operation) the counters (Fig.1), it is possible to obtain the aggregate number of unique elements without counting the same device repeatedly.

With the exception of Scalable Bloom Filters and RIA-DC sketches, all the techniques presented here have the ability to merge, and therefore will not count the same device more than once in aggregate counts. Scalable Bloom Filters lack the ability to merge because their size varies dynamically with the number of unique elements, therefore we cannot guarantee that the same unique device will set the same positions for different filters. RIA-DC sketches might not provide accurate aggregate results since the estimator considers there is no overlap of elements between the different counters.

Aggregation is also important as a mean for answering time related questions like "how many different people were seen in a given gate in the last 24 hours ?" or "what was the number of visitors of an amusement park during visitors peak hours?". To answer these types of questions, it must be possible to distinguish/segment counter readings over time. This can by accomplished by sensor nodes periodically making a copy of their counters followed by a reset. Those copies will keep the information about the unique devices sighted during a certain time period. For example, considering that the rate at which counters

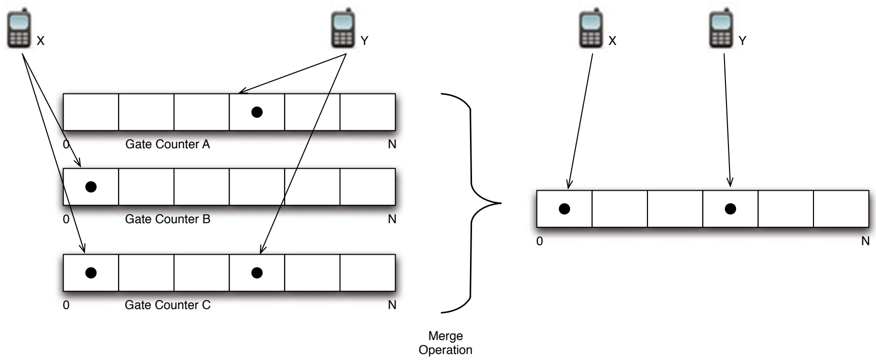


Fig. 1. Merge Operation

are saved and reset would be 1 hour, the former question could be answered by merging the 24 last saved counters. To answer the latter we would need to merge the copies made during peak hours at the various nodes in the park.

To save some space we can use different time granularities, for instance, we can merge all unique counters saved during a day and obtain the aggregate count for the day, merge the counters from the last 7 days and get the aggregate count for the week, and so forth. We just need to keep in mind that because of the merge restrictions, the size of the counter that stores the unique number of devices sighted during an hour has to big enough to fit the number or unique devices seen during the entire week.

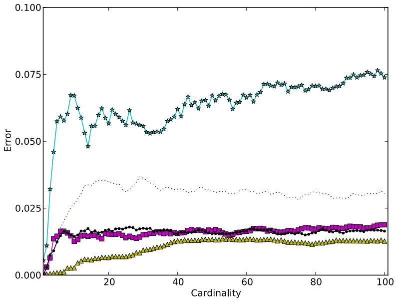
As we can see both *time segmentation* and *aggregate counting* are in fact variations of the same problem, which can only be solved with techniques that support merging.

3.2 Analysis

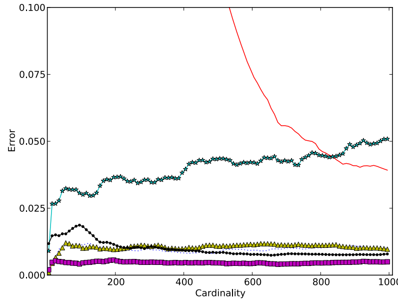
Using the results from our benchmark[1] shown in figures 2, 3(a) and 3(b) and having explained the different criteria we can now analyze each one of the techniques.

- **Standard and Scalable Bloom Filters** provide good accuracy for all the scenarios presented in Fig. 2, never surpassing the stipulated relative error. However, they are the most expensive techniques regarding processing time and the number of bits required per element.
- **LogLog and HyperLogLog Sketches** have very low accuracy for small cardinalities, that is the reason they are not visible in Fig. 2(a). Apart from this issue, both these techniques are the best suited for large cardinalities

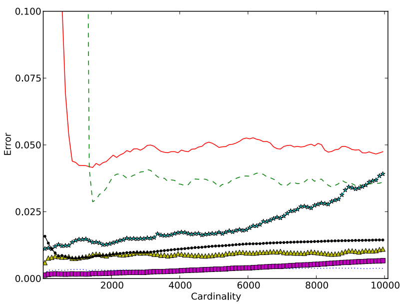
[1] Our Benchmark was built in Python, including the implementation of the several algorithms, with the exception of Bloom Filters where we used Jay Baird's implementation.



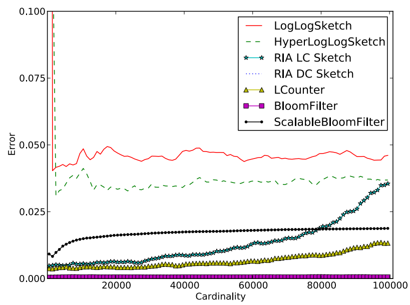
(a) Upper bound 10^2



(b) Upper bound 10^3

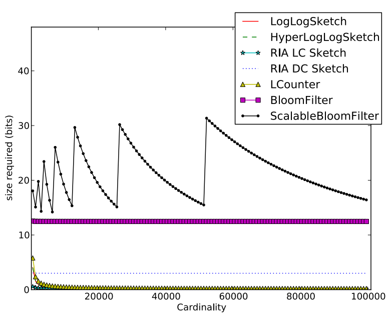


(c) Upper bound 10^4

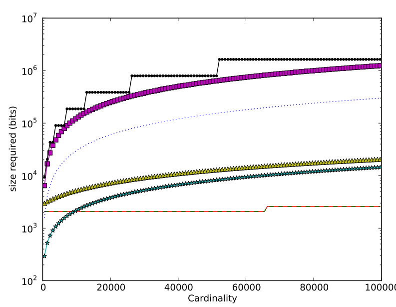


(d) Upper bound 10^5

Fig. 2. Relative error of the several techniques using various upper bounds



(a) Number of bits per element used by each technique at different cardinalities



(b) Size of the several techniques for different cardinalities

Fig. 3. Size benchmarks

due to their logarithmic growth in size (for cardinalities above 10000 these are the techniques that require less space). Between the two, HyperLogLog sketches have the advantage of being more accurate and of having a smaller variability.

– **Linear Counting Sketches** have the same good all around accuracy as Bloom Filters spending only a little more memory than LogLog sketches, therefore drawing the best from each technique.
– **RIA-LC and RIA-DC Sketches**, being based on Linear Counting Sketches, it is no surprise these techniques have good accuracy results. Furthermore and like their ancestor, they also achieve good results in the bits per element ratio.

Taking into account all that has been said, there are a few conclusions to be drawn. For scenarios where we cannot make assumptions on the maximum number of elements to count, we have to use Scalable Bloom Filters. For scenarios where there is a big discrepancy between the cardinalities of different counters and where the existence of repeated elements outside each counter can be neglected, RIA-DC Sketches are probably the best choice. For scenarios with very large cardinalities HyperLogLog Sketches are probably the correct choice since they are the most space efficient technique. Considering the expected most common scenarios, where we need accurate aggregate counts and where there is the possibility of counters with low cardinalities, the choice falls either within Linear Counting Sketches or RIA-LC Sketches. We prefer the latter since it is a slightly simplified version of the former.

As a final remark, we should emphasize the fact that all the techniques discussed in this article are more efficient (check Fig.3(a)) in terms of space than storing each unique device MAC address(48 bits).

4 Conclusion

Bluetooth devices are pervasive in most societies and the number of unique Bluetooth sightings is an adequate proxy for the number of actual individuals present. The trivial approach of collecting and counting the set of detected Bluetooth MAC addresses is not adequate, in most settings, both in terms of privacy concerns, system scalability and adequacy to devices with limited memory.

In this article we described and benchmarked a set of stochastic summarizing techniques that can be applied to the gate counting problem. By using these techniques our approach ensures the privacy of the users since Gate Counters don't store any extra raw information, i.e, the raw information that they keep at any given moment is also present in the Bluetooth network.

Furthermore, the analysis of these techniques and their trade-offs should help to determine the most adequate solutions for a specific gate counting scenario. We also hope to have motivated the community to the relevant role of stochastic counting techniques in privacy preserving gate counting.

Acknowledgments. This research has received funding from FCT under the Carnegie Mellon - Portugal agreement. Project Wesp (Grant CMU-PT/SE/028/2008).

References

1. Almeida, P.S., Baquero, C., Preguiça, N.M., Hutchison, D.: Scalable bloom filters. Information Processing Letters 101, 255–261 (2007)
2. Bloom, B.: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM 13(7), 422–426 (1970)
3. Durand, M., Flajolet, P.: Loglog Counting of Large Cardinalities. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 605–617. Springer, Heidelberg (2003)
4. Fan, Y.C., Chen, A.L.P.: Efficient and robust schemes for sensor data aggregation based on linear counting. IEEE Trans. Parallel Distrib. Syst. 21, 1675–1691 (2010)
5. Fan, Y., Chen, A.: Efficient and robust sensor data aggregation using linear counting sketches. In: IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2008, pp. 1–12 (2008)
6. Flajolet, P., Nigel Martin, G.: Probabilistic counting algorithms for data base applications. Journal of Computer and System Sciences 31(2), 182–209 (1985)
7. Éric Fusy, G.O., Meunier, F.: Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. In: AofA 2007: Proceedings of the 2007 International Conference on Analysis of Algorithms (2007)
8. O'Neill, E., Kostakos, V., Kindberg, T., Schiek, A.F.g., Penn, A., Fraser, D.S., Jones, T.: Instrumenting the City: Developing Methods for Observing and Understanding the Digital Cityscape. In: Dourish, P., Friday, A. (eds.) UbiComp 2006. LNCS, vol. 4206, pp. 315–332. Springer, Heidelberg (2006)
9. Whang, K., Vander-Zanden, B., Taylor, H.: A linear-time probabilistic counting algorithm for database applications. ACM Transactions on Database Systems (TODS) 15(2), 229 (1990)

Touristic Valorisation: A Niche of Interest for Multimodal Information Exploitation

Davide Di Pasquale, Nicola Maiellaro, Marco Padula, and Paolo L. Scala

ITC-CNR, Istituto per le Tecnologie della Costruzione,
Consiglio Nazionale delle Ricerche, Italy
{dipasquale,maiellaro,padula,scala}@itc.cnr.it

Abstract. The aim of this work is the identification of proper tools for the multimodal delivery of information services on various types of terminals, in the context of a valorisation project for the National Park of Alta Murgia, Italy. An open source approach is proposed to the development of technological solutions that fit diverse aspects of the stakeholder's needs, leveraging the dissemination of ICT resources and personal telecommunication terminals able to natively support the multimodal delivery of informative contents.

Keywords: multi-modal communication, tourist heritage promotion, social applications.

1 Background

The increasing availability and accessibility of information in various fields of human activity and the dissemination of personal telecommunication terminals able to natively support a multimodal delivery of contents is the sign of the last decades and, in perspective, a trend in rise. The sharing and participation prerogatives of Web 2.0, is changing how people perceive, access and use the information content offered by content providers: these trends should be exploited to increase the added-value services available in many realities, both from the point of view of private citizens, business and public administrations. This scenario is pushing operators of facilities addressed to a wide audience of users to reformulate their information offer, updating the techniques for services delivery by implementing new technologies and methods of communication. This is happening in many fields: in the finance context, for example, with the development of home-banking and home-trading services, by which it is made available to the user the opportunity to remotely interact with several banking services through telephone, personal computers or mobile terminals. Many public administrations, on the other hand, are offering online services to the citizens for accessing and manipulating information about their rights and duties without queuing in public offices. Public and private medical structures are offering the possibility to access personal information about health care practice (exams booking and analysis results collection) through the use of informatics devices with dedicated equipment for security management (e.g. smart cards reader for secure identification and information storage).

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 544–552, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Application Scenario

This work is contextualized into a project aimed at the valorisation of the National Park of Alta Murgia, Italy. The Park is one of the most extended among national and continental parks, established with d.P.R. 10 March 2004 (G.U.R.I. n.152 of 01 July 2004), expanded about 68 thousands hectares, is included in the Nature Site 2000 SIC/ZPS IT9120007 "Murgia Alta" (about 125.000 hectares). Many characteristic habitats are specific of this Park: the Grasslands on calcareous substrate (Festuca Brometalia) and the grass and local plants (Thero-Brachypodietea) steppe routes, as well as rocky crests, dolines, sweaty hills, sink-holes, karst grottoes, steep-descent scarps, depressions, extended natural pastures and cultivations, pine forests and oak woods. Here, the perennial action of nature is linked with the thousand year action of man which built fortified farmhouses (the so called *masserie*), enclosures with sheep-folds, cisterns, snow-warehouses, little churches, stone heaps hemmed in an infinite network of dry-stone walls. Many tombs carved in the stone discovered in different archaeological sites, a human skeleton perfectly kept unaltered in one of the karst cavity in the surrounding of Altamura and the foot prints of ancient and huge reptiles impressed on the plateau of some spent stone caves are all the confirmations of some historic significance. The long and slow man's action has contributed to build the Murgia landscape, one of the most remarkable landscapes of the Mediterranean area.

The National Park, being an intrinsic synthesis of different aspects in theme of naturalistic heritage, social, touristic and economical interests and environmental protection, represents an application scenario that can benefit by the implementation of multi-modal techniques for the delivery of information and services: the range of services linked to that variety of interests and therefore users, in fact, should be differentiated both in content and access modality on the basis of the targeted audience.

The Administration of this National Park identified the following main needs and proper actions for the enhancement and exploitation of the Park's resources at large:

- promoting the interaction with the community, both local and tourist;
- enabling the remote fruition, even to promote the park in the national and international tourist circuits;
- allowing the sharing of a wide range of information through interactive maps, possibly down-loadable to portable devices;
- increasing the range of environmental education programs;
- supporting slow and Eco-sustainable mobility.

All these aspects can be properly addressed adopting a technological approach that exploits, for example, the ultimate personal telecommunication devices and their multimedia capabilities to improve the users' interaction with the Park's available resources and services. This approach can aid the achieving of a variety of objectives of public utility, like:

- increasing and improving the communication from the Park to the end users;
- sharing of the stock of knowledge gathered over time;
- diversification of informative offer supplied to users on the basis of their interests (historical-architectural and cultural heritage, biodiversity, geology, hydro-geological protection practices, speleological activities);

- increasing visibility of the local food farming resources and promotion of local products;
- the promotion of environmental education programmes aimed to scholar ages and differentiated by age;
- development and dissemination of knowledge resources already existing in the area, such as the network of libraries and information points, updating the modality of access to existing services;
- exploiting the benefits obtained from managing Web-based geographic information system;
- easing the preparation of emergency and risk plans to local public managers and administrators;
- consolidation of local knowledge and implementation of monitoring plans of natural or historic heritage resources;
- improving information access for individuals, governments and businesses.

We believe that such multifaceted objectives are fully supported by the availability of personal technological devices which can be easily customized and integrated with extra hardware (e.g. GPS antennas) and/or specific software applications in order to allow the exploitation of offered solutions with interactions modalities the user is accustomed to and suited to the aim he/she has in mind.

3 Aim of the Work

Aim of this work is the identification of proper tools for the exploitation of recent techniques and solutions for the multimodal delivery of information services on various types of terminals, exemplifying the possible profit achievable with their implementation, either adapting existing services or developing new ones, in the context of a valorisation project for the National Park of Alta Murgia [1-4].

The multimodal approach that characterize this proposed set of tools finds expression in the possible modes of fruition of the informative contents that will be made available. The multimedia and multimodal capabilities of modern devices, personal computers and mobile terminals, intrinsically support and ease this approach, stimulating the implementation of applications that exploits the up-to-date paradigms of the human-machine interaction.

4 Methodology

The proposed methodology for the realization of this set of tools can be summarized into the following steps:

- identification of the different types of services, their user target and technological media for the delivery. A schematic summary of proposed services is shown in table 1;
- choice of a methodology for the design of a customizable development environment for creating applications for the delivery of multimodal information content;

- determination of the proper development platform that fits the needs of individualized content and customized modal delivery;
- analysis of pre-existing information available at local level;
- migration of the contents into the new platform or its fitting to the new specifications;
- implementation of the management interfaces and user interfaces;

5 Proposed Tools and Technical Solutions

A significant portion of information necessary and available for the type of scenario here depicted is, obviously, intimately connected with the territory and must therefore be made available as geo-referenced data. The information delivery tools must therefore be able to manage the entire informative workflow providing the characteristic spatial metadata in order to ease the customization of service, for example on the basis of the user's location. The main tools proposed to deal with the listed topics are:

- **Web-GIS:** Spatial information requires for its treatment and analyses special software tools, namely geographic information systems (GIS) [5, 6]. GIS management tools have been traditionally developed as desktop standalone applications and have become an important part of a variety of information systems used for supporting decision-making processes in business, public administration, and personal matters. Among the most significant Open Source software applications, should be cited: *gvSIG* [7], a local and remote map viewer (2D and 3D) with support for the OGC standards; *GRASS* [8], developed for Unix platform, it's a mature software with support for vector geometries and raster data, managing of volumetric pixel (voxels) in 3D rendering and APIs for Python, Perl and PHP; Quantum GIS or *QGis* [9] runs on Linux, Unix, Mac OSX, and Windows, supports vector, raster and database formats and is coded upon the QT library by Nokia. It supports all OGC-compliant services like WMS , WMS-C (Tile cache), WFS and WFS-T. Among commercial desktop GIS the most diffuse software are: ESRI's *ArcView* [10] features map authoring with templates, spatial query with query-building tools, basic modeling and analysis with custom reports generation, simple feature editing and data integration; *Bentley Map* [11] natively manages Oracle Spatial to store and edit all types of spatial data; Intergraph's GeoMedia products suite [12] is a set of integrated applications that provide a set of analytic and editing tools.

  Web-GIS are an implementation of the GIS approach to geo-referenced data applied to web browsing terminals, by mean of web based interfaces. The tool provides access to a wide range of services, like naturalistic and tourist trails planning and personal routing, exploitation of mobile maps with GPS Points of Interest, contextualized information personalized on the basis of proximity, thematic area (culinary, geological, historical, etc.) or sustainability (Eco-sustainable/slow mobility); The proposed tool is a system based, server side, on MapServer [13] an Open Source geographic data rendering engine written in C. The tool supports popular scripting and development environments like PHP, Python, Perl, Ruby, Java, and .NET and is cross-platform (Linux, Windows, Mac OS X, Solaris). It supports many Open Geospatial Consortium standards like WMS (client/server), non-transactional WFS (client/server), WMC, WCS, Filter Encoding, SLD, GML, SOS, OM and manages a multitude of raster and vector data formats (TIFF/GeoTIFF, EPPL7, and many others via GDAL, ESRI shape files, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL and many others via OGR). On client side the proposed solution is the open source JavaScript library OpenLayers [14] released under a BSD-style license. With its implementation inside a web page it is possible to display map tiles and markers loaded from diverse sources, both from local Mapserver engine and external resources (e.g. GoogleMaps, LiveMaps) allowing the exploitation of geographic information available as proprietary format files (e.g. ESRI shape files) as well as open access files (e.g. OpenGIS Web Map Service Interface Standard, WMS [15]) directly inside a web browser.

The possibility of offline usage of available information is given via the download of customized map files to be used within Navit [16], an open source car navigation system with routing engine which supports mobile devices. It supports the OpenStreetMaps format for maps and various formats for POI and tracks (gpx, cvs, plain text). For the use inside the Park, the webGIS web interface can provide proper maps and POIs to be downloaded on user request. The proposed tools represent the core facilities for the implementation of varied services, for instance the visits planning, the naturalistic trails, mobility, accessibility and tourist promotion services (see Table 1);

- **local social network,** an approach to social networks based on customizable themes: communities are built on the fly on the basis of GPS location (for example users who are within 1 km or those who are in a particular place) or of thematic relevance (interest in gastronomic, cultural or naturalistic topics), or of users' nationality. This facility provides a tool for the use and sharing of user-level produced content;

  The proposed tool makes use of GPS capabilities of recent mobile devices joined with the compliance of recent mobile browsers (e.g. Opera Mobile 11 [17]) to the W3C Geo-location API Specification [18], which allow to query mobile hardware data (like GPS data or GSM network information) client-side and to pass it to the server. A customized version of the open source chat platform Ajax-Chat [19] is proposed to manage users login into chat channels on the basis of their geo-location or other preferences (interests, etc.). AJAX Chat is a fully customizable web chat implemented in JavaScript, PHP and MySQL which requires PHP 4+ and MySQL 4+ (Server-Side), JavaScript and enabled Cookies (Client-Side);

- **webcams,** an intranet network of static and live webcams (visible/infrared) providing a monitoring facility with application to fire prevention and risk management for local public administrations. The tool can also be made available for the promotion of tourist and cultural attractions. While static webcams images can be embedded in any web page, eventually temporising image updates with client side scripts or java applets, a live webcam stream needs a streaming server to deliver video data flow. The proposed streaming server solution is the open sourced Darwin Streaming Server (DSS [20]). Developed by Apple , it is the open source equivalent of QuickTime Streaming Server, and is based on its code, running on Mac OS X, Linux, FreeBSD, Solaris, Tru64 Unix, Mac OS 9 and Windows. It is a fully featured RTSP/RTP media streaming server capable of streaming a variety of media types including H.264/MPEG-4 AVC, MPEG-4 Part 2and 3GP. This tool can support the enabling of monitoring and tourist promotion services, implementing different access policies to the available contents;

- **users video content creation,** a facility that allows users to share self-generated video contents through the park's portal. This can be achieved with a video sharing platform where users can upload and share contents. The solution proposed is MediaCore [21], a platform built in Python and released under the open source GPL license. It runs on Linux O.S. and requires Python 2.5+ and MySQL 5.0+; it is completely themeable with CSS and XHTML or through the administrative interface, being so adaptable and embeddable to the Park's portal. The platform supports many media formats like FLV, M4A or MP3 and can serve video to both desktop browsers and mobile devices. It supports HTML5, falling back to Flash where browsers do not support newer standards. The possibility to programmatically query the media library is given to developers through the support of JSON APIs, making it possible to embed active links in other websites. It natively supports user generated content via the upload interface: the administrators are notified and can choose to review and approve new content. It allows to comment and share video or audio with friends through the most popular social network and has built-in support for the visually impaired through closed-captioning and audio descriptions, being compliant to the United States' laws on accessibility;
- **Virtual Tours,** a set of immersive panoramic views of sites of particular interest (naturalistic, historical, socio-economical,...) that allow remote visualization for personal or tourist promotion purposes, enabling also virtual accessibility of locations non otherwise accessible to impaired users. The proposed platform for the visualization of full immersive panoramas is the flash based software Flash Panorama Player [22], which is fully cross-platform. Supports jpg, png and gif image file formats, cube faces panorama format or cube stripes, is optimized for full screen with featured anti-aliasing and can be embedded in any HTML or Flash site as well as it can be used to create both online and offline applications. This tool can support the visits planning and accessibility services.

6 Multimodal Interaction: A Use Case

The interaction with the proposed tools can take advantage from the exploitation of novel HCI paradigms implemented on latest devices by leveraging the quantity of sensor and dedicated hardware present on most of them. The interaction with the web-GIS application, for example, can be carried out exploiting the touch panel of the mobile device to pan and zoom the map or open informative geo-localized marks on it; another modality of interaction leverages the vocal recognition and speech synthesis capabilities of the Navit platform, supporting hand-free operations during navigation and letting, for example, bikers follow bike routes or listening to POI descriptions without holding the device. As a use case an example of user experience in the exploitation of the proposed tools is hereafter described.

Mr. John Doe is planning a journey in Italy and is going to visit the Puglia region. Surfing the Internet he finds the homepage of The National Park of Alta Murgia and is attracted by the landscapes shown in the webcam section of the web site. A set of immersive virtual tours captures his attention with a variety of naturalistic environments and interesting places; he finds also views of archaeological sites and decides to further deepen this topic downloading tourist guides and other information about the ancient civilization of the region. In the Park's Web site the location of tourist points of interest (POI) are shown using a thematic map where each featured element is geo-located and presented as a clickable balloon with description and further links. Accessing the shown-features selector of the map, Mr. Doe can switch between different categories of POI beyond the archaeological theme, for instance naturalistic sites, restaurants, tourist receptivity and transport facilities. This very intuitive interface allows him to evaluate different solutions for his accommodation, optimizing the choice of the location on the basis of his interests. Mr. Doe loves bicycle and is very pleased to find a map theme with suggested bike trails inside the park: he downloads the suggested local navigation application for his mobile device in order to be able to follow the routes in case of connectivity issues once in the outback. Using the links to tourist useful information, Mr. Doe easily plans his trip and books planes and accommodation for him and his family. Once Arrived in the Murgia, he exploits the downloaded application as a personal routing facility that ease him finding the hotel and any other POI in the surroundings. Accessing the National Park's website with his GPS-enabled mobile device he gives a try to the local social network (LSN) service and after having set up his profile with nationality and main interests for the ongoing holydays - bike trails, archaeology and gastronomy – the service highlights on a map other current users with similar characteristics in the surroundings, allowing instant messaging sessions with them. Using this facility he finds a group of English cyclists who are planning a bike trail in the park for the following day and joins them with his family at a nearby restaurant that gives special offers for the users of the LSN service. The following day he leaves for the bike trip with the new friends while his wife, lightly impaired for a knee wound, prefers to visit a local museum. During the ride he follows the route on his mobile device with the hand-free navigation featured by the application and proposes a short detour to reach an archaeological site showed in the vicinity of their current position on the navigator map. That evening, once back in the hotel, he shares with his wife the otherwise precluded to her experience of visiting that ancient ruins, showing her the immersive environment offered by the virtual tour section of the Park's web site. He decides to share with other people the highlights of the day trip uploading the video clips he recorded during the bike ride on the media sharing platform hosted in the Park's portal, emailing then the resulting URLs to his friends. It's now late in the night and all tourist offices are closed, but he can access a widespread set of information on local resources to plan his activities for the upcoming day in the National Park of Alta Murgia.

7 Conclusions

In this paper a set of tools for the multimodal delivery of information services on various types of terminals are proposed and described in the context of a tourist valorisation project for the National Park of Alta Murgia, in Italy. The proposed technical solutions are based on open source free software and leverage the multimedia capabilities of modern personal devices along with their native multi modal interaction characteristics.

The customized information delivery services oriented to the exploitation of environmental and tourist resources is identified as a niche of interest for the development of software tools which need intuitive, immediate, multiple and coexisting paradigms of interaction with the end user.

Table 1. Summary of proposed services

Users	Service	Terminal	Notes
tourist	*visits planning*	multimedia PC	Virtual tours, webcams, receptivity, downloadable documents, tourist information.
tourist	*naturalistic trails*	PC	maps, info and in-depth links.
tourist	*naturalistic trails*	mobile device	maps, GPS POI , context information.
tourist	*mobility*	mobile device	**personal routing**: contextual; thematic (culinary, geological, historical, etc.); Eco-sustainable.
tourist	*accessibility*	multimedia PC	Virtual tours and webcams enabling accessibility for impaired users.
tourist	*local social network*	mobile device	on-the-fly creation of temporary communities based on: - GPS or location (i.e. those who are within 1 km or user who is in a specific place; - thematic interest (gastronomic, cultural, naturalistic...); - user nationality; Aimed to the creation and sharing of user-defined knowledge.
local administrators	*monitoring*	intranet/internet	network webcam (visible / infrared) for monitoring fire hazards and promote tourist sites.
local business	*advertising*	users' mobile terminals	geo-localized advertising.
local administrators	*touristic promotion*	users' mobile terminals	promotion of cultural attractions via advertising, virtual tours and webcams.

References

1. Costabile, M.F., Fogli, D., Marcante, A., Mussio, A., Parasiliti Provenza, L., Piccinno, A.: Designing customized and tailorable visual interactive systems. International Journal of Software Engineering and Knowledge Engineering 18(3), 305–325 (2008) issn: 0218-1940
2. Barricelli, B.R., Maiellaro, N., Padula, M., Scala, P.L.: Realizzazione di un framework collaborativo per l'authoring di informazioni turistiche ed ambientali e per la loro pubblicazione in rete. In: Maiellaro, N. (ed.) Albania: Conoscere, Comunicare, Condividere, Bari, Italy (2008)
3. Barricelli, B.R., Maiellaro, N., Padula, M., Scala, P.L.: A collaborative system for environment and tourism information authoring and Web publishing: an institutional case study. In: ICIW 2008, Athens, Greece, June 8-13 (2008)
4. Barricelli, B.R., Maiellaro, N., Padula, M., Scala, P.L.: Strumenti open source per la pubblicazione in rete di informazioni turistiche personalizzate. In: ASITA 2008, L'Aquila, Italy, October 22 (2008)
5. Konecny, G.: Geoinformation: Remote Sensing, Photogrammetry and Geographic Information Systems, 1st edn. Taylor & Francis, London (2003)

6. Longley, P.A.: Geographic Information Systems and Science, 1st edn. John Wiley & Sons, Chichester (2001)
7. GVSIG, http://www.gvsig.org/
8. GRASS, http://grass.fbk.eu/
9. Open Source Geospatial Foundation. QuantumGIS, http://www.qgis.org/
10. ESRI. ArcView, http://www.esri.com/software/arcgis/arcview
11. Bentley Systems Inc. Bentley Map, http://www.bentley.com/en-US/Products/Bentley+Map/Product-Overview.htm
12. GeoMedia, http://www.intergraph.com/sgi/products/productFamily.aspx?family=10
13. The MapServer site, http://mapserver.org/
14. The OpenLayer site, http://trac.osgeo.org/openlayers/
15. The Open Geospatial Consortium site, http://www.opengeospatial.org/standards
16. The Navit site, http://www.navit-project.org/
17. The Opera site, http://www.opera.com
18. The W3C Geolocation API Specification, http://www.w3.org/TR/geolocation-API/
19. The Ajax-Chat Open Source Web Chat project site, http://sourceforge.net/projects/ajax-chat/
20. The Darwin Streaming Server site, http://dss.macosforge.org/
21. The MediaCore website, http://mediacore.com/
22. The Flash Panorama Player site, http://flashpanoramas.com/

Assessing Feedback for Indirect Shared Interaction with Public Displays

Jorge C.S. Cardoso[1] and Rui José[2]

[1] Escola das Artes / CITAR – Univ. Católica Portuguesa, 4169-005 Porto, Portugal
jorgecardoso@ieee.org
[2] DSI – Univ. do Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
rui@dsi.uminho.pt

Abstract. Interaction is a key element in turning public displays into a platform for social interaction, making them more engaging and valuable. However, interactive features are still rare in public displays, due to the lack of generalised abstractions for incorporating interactivity. In our work, we explore to what extent the concept of interaction widget, which was so successful on desktop computers, could also be used as an abstraction for remote, shared interaction with public displays. A particular challenge is presenting input feedback in this shared, multi-user, and indirect interaction setting. In this paper, we present a study on the feedback mechanisms of these widgets, to determine if users are able to understand the results of interactions in single and multi-user settings. We have evaluated three feedback mechanisms and the results indicate that the general mechanism provides an appropriate sense of what is happening and could in fact provide general awareness of the interaction alternatives and current status, even in multi-user scenarios.

Keywords: interactive public displays, widgets, remote interaction.

1 Introduction

Public digital displays are becoming increasingly ubiquitous artefacts in public and semi-public spaces. Interactive public displays are particularly well suited for social interactions, because they are part of a share physical space and naturally provide a shared digital medium that anyone can see and use. Most of them, however, do not support any type of interactive feature, despite interaction being clearly regarded as a key element in making them more engaging and valuable. A key reason behind this apparent paradox is the lack of generalised abstractions for supporting interaction, which means that each new application developer will need to create his or her own approach for dealing with a particular interaction objective using a particular interaction modality. The consequence is too much development work outside the core application functionality to support even basic forms of interaction and an obvious waste of development effort, potentially leading to poor designs. This is also a problem for users, as they need to deal with inconsistent interaction models across different displays, representing a major obstacle to the emergence of any expectations and practices regarding interaction with public displays. Given that interaction is

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 553–561, 2011.
© Springer-Verlag Berlin Heidelberg 2011

considerably more limited than it is in desktop environments, and thus there are not many interface exploration possibilities, it is even more important to be able to convey the interaction affordances of the display in a way that people can easily understand.

Early desktop computer programmers had to make a similar effort to support their interaction with users. This was quickly recognised as a problem and addressed with the emergence of reusable high-level interaction abstractions, including widgets, which provided consistent interaction experiences to users and shielded application developers from low-level interaction details [1].

In our work, we are exploring to what extent the concept of interaction widget, which was so successful on the desktop, could also be used as an abstraction for some forms of interaction in public displays. While this may eventually lead to totally new widgets, we started by investigating to what extent some well-known desktop widgets could migrate to this new usage context. Well-known widgets are particularly attractive as an entry strategy because people are already familiar with them and may easily recognise them as interaction opportunities.

We are considering specifically the case in which multiple people may be concurrently interacting with a shared display using remote devices, such as mobile phones or portable computers. Multiple interaction techniques may be used, e.g. Bluetooth or SMS, and a key design goal is that widgets should be as much as possible independent from specific input channels. Creating widgets for this type of indirect interaction model raises two fundamental issues that are not present in traditional widgets. The first is that the direct manipulation paradigm does not apply because people are not acting directly on the widget. Feedback, or at least the type of feedback that is normally associated with widgets, is either non-existent, or would never be as immediate as the expectations that people have when interacting with traditional widgets under direct manipulation paradigms. A second issue is that multiple people may be acting concurrently on the same widget and therefore the type of feedback given may need to indicate to whom, from the potentially multiple people interacting at a particular moment, a specific feedback is directed to. The main implication of these issues is the need to create a specific feedback model that enables people to understand the effect of their actions and possibly the actions of others.

In this paper, we focused on the feedback for these widgets and we evaluated three different mechanisms for presenting visual feedback about input events directly on the public display. Results indicate that users are generally able to understand the feedback to their input and that of other users, giving us confidence that the proposed feedback model is appropriate and can be further explored.

2 Related Work

Widgets have been created for many specific uses [2, 3], but none for general-purpose public displays. For example, Rohs [4] has implemented a set of widgets for visual marker-based interaction that allows users to activate actions or select options encoded in a visual marker and send it via SMS (using a custom mobile application). In this case the visual marker encodes the type (menu, radio or check button list, sliders, etc.) and layout (vertical or horizontal menu, number of options, etc.) of the widget, so that the mobile phone application can immediately superimpose graphical information about the currently selected item or value. However, these widgets are limited to camera-equipped mobile phones and were not developed with multi-user interaction in mind. This approach requires users to be relatively close to the display in order to be able to use the mobile phone's camera with enough quality (or the use of larger visual markers); also, system feedback is shown only on the personal mobile phone and not on the public display limiting other user's understanding of what is happening in the display application.

Dearman & Truong [5] developed Bluetone: a widget that is activated through dualtone multi-frequency (DTMF) over Bluetooth. Users interact with an application by changing the Bluetooth name of their device to a system command, wait for the display to pair with the user's phone as an audio gateway, and then pressing the keys on the keypad of their phone. Bluetone supports several users, being limited only by the Bluetooth protocol (7 connections). This widget is limited to the DTMF interaction mechanism, and has been developed for an environment where a single application executes at a time; graphically, it consists of a single widget that encapsulates all the interactive features of the application. Also, Bluetone does not directly address how feedback is presented in a multi-user setting.

3 PuReWidgets

PuReWidgets is an initial implementation of a widget toolkit for public display applications. PuReWidgets keeps the visual appearance and basic affordances of their desktop counterparts, but adapts them to address the specificities of public display interaction. Like traditional WIMP widgets, these widgets are easily identifiable as possible actions to be performed on the public display and provide applications and users with a single abstraction for multiple input mechanisms. Unlike traditional widgets, they are not based on direct manipulation and multiple people can use them simultaneously.

The association between interaction events and specific widgets is achieved through a generic referencing scheme, which consists of automatically generating unique textual references whenever an application adds an instance of a widget to the interface. This reference is graphically shown on the widget instance (e.g., "btn1" in the button of Fig. 1) and it serves as an addressing mechanism for users and for the display system. Whenever a user enters the reference code in an input command, the display system routes the command to the appropriate widget instance. This referencing scheme is common for many SMS services, so most users will be familiar with it. Although, in this study, we focused on SMS as the input mechanism, the referencing scheme could be used in many other text input mechanisms such as Bluetooth naming, email, instant messaging but also by mechanisms such as visual markers or even custom mobile applications.

In our initial set, we included five of the most common input controls in traditional GUI interfaces: button, checkbox, textbox and two variants of a listbox. Figure 1 shows the initial widget set. The reference code is shown between brackets in all widgets. Widgets with several options, such as listboxes, have several reference codes assigned to them so that users may address each of the available options. A widget may be enabled or disabled. Widgets should generally be enabled, but applications may choose to disabled a widget temporarily, for example, to signal that a particular operation is not available at the moment. Disabled widgets use the traditional "greyed out" style.

Button	Checkbox	Textbox	Listbox Single	Listbox Multiple
Play (btn1)	CheckMe (chk1)	Submit a tag (txt1)	option 1 (lst1) option 2 (lst2) option 3 (lst3)	option 1 (lst1) option 2 (lst2) option 3 (lst3)

Fig. 1. Initial widget set showing the enabled state

The PuReWidgets feedback model allows multiple users to interact concurrently, while being able to correctly display feedback to each user's input in an unambiguous manner. Feedback is shown directly on the public display, next to the widget that received the input. The user identification (taken from the input channel, e.g. phone number) is included in the feedback information, masked in a way that enables the sender to recognise his input in the feedback, but other users will only be able to distinguish different users, not to identify them. This shared feedback on the public display, next to the widget (or inside it), instead of, for example, directing it to the user's mobile device, is independent of the input mechanism: it does not assume that a return channel is available. It also provides a simple awareness mechanism that conveys information about other people's interactions. This should serve to entice more people to interact [6] and also to enable viewers to make sense of the display behaviour.

Feedback is seen here as system level feedback, i.e., it translates an acknowledgement that a particular input has been received and how it has been interpreted, although not necessarily accepted. Input can either be accepted by the widget, or rejected, but it always generates feedback to the user. Accepted input is passed on to the application, translated into an application event, while rejected input does not result in any application event. Input can be rejected for two reasons: the widget could be disabled; or the user made an error in a parameter needed by the widget (for example, forgetting to enter a string in the texbox widget).

3.1 Usage Scenario

The initial widget set provides a large range of possible interactive features for public displays. Although their exact graphical appearance may change considerable, their basic affordances can be used for a variety of situations. Buttons for “liking” items on social or even regular webpages are ubiquitous now and can also have a big impact on public display content. Textboxes can be used to signal the possibility of entering comments, perhaps filtered before they show up next to a content item on a public display. Listboxes are the building blocks for polls and questionnaires that can be used by public displays to collect users’ preferences and generating discussion about current national or local hot topics. Checkboxes are an easy way to let users rapidly configure a public profile by checking or un-checking profile items.

The following scenario gives an example of how some of these widgets might be useful for public display interaction:

Sophia has just entered her university's main hall and is looking for a place to sit down and wait for her colleagues -- the first class is only due in 15 minutes. As she sits down, she looks at the large display across the hall noticing it is showing a list of local news related to her school. One of the entries catches her eye -- it's about Adam, a colleague on the robotics class, which has won the national robot dancing contest. Sofia notices a button next to the news entry's header and recognises it from her favourite social website: is a ``like'' button with three letters underneath. The instructions on the top of the display tell her how to interact so she fetches her mobile phone and sends a text message to the number on the instructions. A few seconds later, text pops up on the button: something resembling a phone number with some digits erased appears, and she recognises it as her own. She knows her ``like'' will increase the news visibility on the school's website and on the display. Adam deserves it! At the same time, a group of students passes by, and one of the girls points to the display and makes some remarks about it. The button’s animation triggered by Sophia's interaction apparently caught the girl's attention.

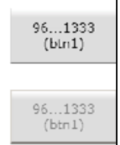
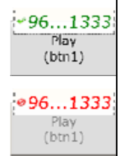
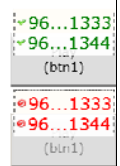
Sophia also notices a vertical bar on the left of the display, showing a list with the last people that interacted with the display. She recognises her phone number as the last entry, and because the display does not have much information about her, there is also a textbox that allows her to enter a nickname. Sophia hears someone calling her name. Looking away from the display she sees one of her classmates signalling her. It's time for classes; filling in the nickname will have to wait.

4 User Study

This study’s main purpose was to allow us to have an initial assessment of the feedback model for PuReWidgets and possible implementations. To assess the effectiveness of the feedback, we evaluated three alternative implementations: Internal, External, and External-Cumulative (described in Table 1). The study was focused on the following research questions:

- Will users be able to identify and understand the feedback to their input?
- Will users be able to identify and understand feedback directed at other users?
- Will there be a difference in error rate between the several feedback mechanisms?
- What will be the preferences of users regarding the feedback mechanisms?

Table 1. The three implementations of the feedback model. The right column shows a button with feedback to an accepted input (top), and with feedback to a rejected input (bottom).

Internal feedback (Int) uses the widget's internal text components and temporarily changes them to show input feedback. Accepted input is shown in the same style as the label. In the case of the listbox, feedback is shown directly in the label of the selected option. Feedback to rejected input is formatted with a strike-through text style and grey colour. This mechanism is the one that requires fewer changes to the visual appearance of the widget because it simply reuses, temporarily, a text component of the widget.	
External feedback (Ext) uses a pop-up panel that is displayed next to the widget (by default, over the widget, slightly raised). In this case, since it is not bound to the existing components of the widget, feedback can use a richer visual style. We use a green "check mark" icon and green text to indicate an accepted input. To indicate rejected input, the style of the feedback is changed to show a red "prohibited" icon and red text.	
External-Cumulative (E-C) is an extension to the external feedback mechanism. In order to facilitate the comparison of multiple input feedbacks, the pop-up panel maintains the previous feedback texts (within a pre-configured period of time) and is incremented with the current feedback text. The style for the accepted and rejected input is the same as the External mechanism. When there is only a single input, the External-Cumulative mechanism behaves in the same way as the External mechanism.	

4.1 Procedure and Tasks

The study was structured in three parts: a first part in which participants were asked to interact with widgets in a single user setting, a second part in which participants were asked to interact with widgets in a multiple user setting, and finally a questionnaire in which users were asked to indicate the best and worst feedback mechanism for each widget. Participants were told about the objectives of the study and were given minimal information about how to use the widgets: they were told that to activate a widget they had to send an SMS message with the reference code that was shown inside the widget. Participants were divided into three separate groups according to the feedback they would be subjected to: Internal, External, and External-Cumulative. For part 1, the group External-Cumulative was incorporated into group External because both mechanisms behave the same way when only feedback to one input is shown. This meant that part 1 had two groups, and parts 2 and 3 had three groups.

Participants and Apparatus. We recruited 24 participants (all daily computer users) from the university campus (8 female, 16 male; ages from 22 to 41). The experiment was done in a room equipped with a 23 inches LCD display. Participants sat at about two meters away from the display and were given a mobile phone to use during the experiment. Interaction was done via SMS.

Part 1 – Single Input. For part 1 of the study, participants were presented with a sequence of five screens with one widget on each screen. Each screen depicted a simple mock-up application to give some context to the interaction, with a single widget placed in it. The application itself did not respond to any interaction, only the widget reacted to the input. Each screen presented a task to be completed by the participant. The five screens were presented in a random order. In two randomly selected screens, the widget was disabled (i.e., any input would fail). Participants were told to complete the presented task (they could perform several trials) for each screen and, after receiving the widget's feedback, were asked: "Did you successfully activate the widget?" We recorded the responses (yes, no, don't know) and number of trials that each participant took before answering. We recorded an error for each wrong response (a "don't know" counted as wrong) and for each extra trial needed.

Part 2 – Multiple Input. In this part of the study we simulated the case were several users would interact with the same widget at the same time. We chose to simulate the interactions of other users instead of performing a multi-user experiment, in order to have full control over the interaction sequence (guaranteeing the occurrence of collisions between the input of multiple users). Participants were presented with the same five screens of part 1 (in a randomised order) and were told that there would be other users interacting at the same time as they were, but not necessarily making the same choices. All the screens were configured so that the widget would become automatically disabled after receiving the first input. Again, participants were told to complete the presented task (they could perform several trials) for each screen. After each screen (after the user has issued the input and seen all feedback), participants were asked: "Did you successfully activate the widget?", "How many other people were interacting with the display?" and "How many of those other people successfully activate the widget?"

Part 3 - Subjective preference. In the final part of the study we showed participants the three alternative ways to display feedback for all the widget types. We asked participants to observe the different feedback mechanisms for as long as they needed and then to indicate the best and the worst mechanism for each widget type. We recorded the participants' answers, allowing them the choice of not selecting any as the best or worst.

5 Results

For part 1 and 2 of the study, we calculated the total number of errors (wrong answers and extra trials) made by each participant over the five screens in each part (shown in the boxplots of Fig. 2). For part 1, a right-tailed t-test was performed to test the hypothesis that the average error is higher in the Int condition than in the Ext condition. The test was significant ($t(22) = 5.87$, $p < 0.0001$), suggesting that users have more difficulty with the Internal mechanism.

For part 2, we did a pairwise comparison of the three conditions using Welch's Two Sample t-test and then applied Bonferroni's correction to the p-values. The result was statistically significant for the Int vs Ext ($p = 0.0134$) and Int vs E-C ($p = 0.0076$) tests and nonsignificant for the Ext vs E-C ($p = 0.6982$) test. As in part 1, this suggests that the participants have significantly more difficulty with the Internal mechanism than with the External or External-Cumulative ones.

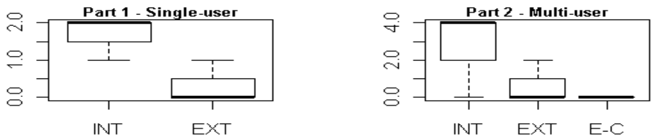


Fig. 2. Boxplots of the total number of errors committed by participants in part 1 and part 2

In part 3 we asked participants to indicate the best and worst feedback mechanisms for each of the widget types. The relative frequency of the occurrence of each mechanism in the "best" and "worst" classifications is shown in Table 2. From these results it is apparent that the External-Cumulative mechanism had the highest positive influence in the "best" classification, and the Internal mechanism resulted in the highest influence for the "worst" classification.

Table 2. Relative frequency of the occurrence of each feedback mechanism in the "best" and "worst" classifications (percentages were rounded to the nearest integer)

	Internal	External	External-Cumulative
Best	28%	6%	66%
Worst	60%	31%	8%

6 Discussion

The results from part 1 and 2 clearly indicate that users perform worse under the Internal feedback mechanism. Participants subjected to the Internal feedback mechanism often expressed confusion about the meaning of the feedback. Another source of frequent errors was telling the number of users who were interacting, in part 2. Participants often missed a feedback event when faced with the Internal mechanism. These results were expected since the graphical cues used in the Internal mechanism are subtler than in the External and more propitious to being unnoticed.

Results do not show a statistically significant performance difference between the External and External-Cumulative mechanisms. However, results from part 3 suggest that users generally prefer the External-Cumulative mechanism (the results also suggest that users generally consider the Internal feedback mechanism to be the worst).

In some widgets, participants were divided about which mechanism was more suited: the Internal or the External-Cumulative. Participants liked the Internal mechanism when applied to textboxes and listboxes. In the textbox case, some participants felt it was more natural to present the feedback (which included the submitted text) inside the textbox, as it would happen if a user were entering text using a keyboard on a desktop computer, resulting in a preference for the Internal mechanism, for textboxes. In the listbox case, some participants liked the fact that the feedback appeared exactly on the options that were selected giving a more direct cue about what options were affected. Still, usage of textboxes and listboxes did not result in a statistically significant difference in performance, which suggests that, although users would prefer a slightly different visual cue, it did not affect how well the information was perceived.

The low error rate of the External and External-Cumulative mechanisms also suggests that the basic idea behind the feedback mechanisms (on screen with users identified by the masked phone number) is understandable and does not present any significant difficulty to users.

7 Conclusion

We have evaluated three different on screen feedback mechanisms for public display widgets in a controlled user study. The performance of the External and External-Cumulative mechanisms indicates that users are able to identify and understand the feedback for their input and for other people's input. The study also showed that users generally prefer the External-Cumulative feedback mechanism, although, for some widget types, there is still room for improvement regarding the feedback.

Future work on this subject will need to evaluate other aspects of the interaction with these widgets. Namely, it is important to study if, in a real setting, these widgets are effectively perceived as interaction opportunities and to what extent a real social context may affect that perception. The number of widgets used by an application, the number of users submitting input simultaneously to the same or different widgets in an application, as well the level of distraction, and the feedback latency are all variables that could affect the performance of this feedback approach in a real setting.

Acknowledgements. The research leading to these results has received funding from FCT under the Carnegie Mellon - Portugal agreement. Project Wesp (Web Security and Privacy (Grant CMU-PT/SE/028/2008). Jorge Cardoso has been supported by “Fundação para a Ciência e Tecnologia” and “Programa Operacional Ciência e Inovação 2010” (POCI 2010), co-funded by the Portuguese Government and European Union by FEDER Program and by “Fundação para a Ciência e Tecnologia” training grant SFRH/BD/47354/2008.

References

1. Swick, R.R., Ackerman, M.S.: The X Toolkit: More Bricks for Building User Interfaces, or Widgets for Hire. In: Proceedings of the Usenix Winter 1988 Conference, pp. 221–228 (1988)
2. Conner, B.D., Snibbe, S.S., Herndon, K.P., Robbins, D.C., Zeleznik, R.C., van Dam, A.: Three-dimensional widgets. In: I3D 1992: Proceedings of the 1992 Symposium on Interactive 3D Graphics, pp. 183–188. ACM, New York (1992)
3. Ryall, K., Esenther, A., Forlines, C., Shen, C., Shipman, S., Morris, M.R., Everitt, K., Vernier, F.: Identity-Differentiating Widgets for Multiuser Interactive Surfaces. IEEE Computer Graphics and Applications 26(5), 56–64 (2006)
4. Rohs, M.: Visual Code Widgets for Marker-Based Interaction. In: ICDCSW 2005: Proceedings of the Fifth International Workshop on Smart Appliances and Wearable Computing, pp. 506–513. IEEE Computer Society, Washington, DC (2005)
5. Dearman, D., Truong, K.N.: BlueTone: a framework for interacting with public displays using dual-tone multi-frequency through bluetooth. In: Ubicomp 2009: Proceedings of the 11th International Conference on Ubiquitous Computing, pp. 97–100. ACM, New York (2009)
6. Brignull, H., Rogers, Y.: Enticing People to Interact with Large Public Displays in Public Spaces. In: Rauterberg, M., Menozzi, M., Wesson, J. (eds.) INTERACT 2003, pp. 17–24. IOS Press (2003)

Toward New Communication Paradigms to Enhance Cognitive and Learning Processes

Maria Chiara Caschera, Alessia D'Andrea, Arianna D'Ulizia,
Fernando Ferri, Patrizia Grifoni, and Tiziana Guzzo

Institute of Research on Population and Social Policies
National Research Council
Via Palestro 32
00185, Rome
{mc.caschera,alessia.dandrea,arianna.dulizia,fernando.ferri,
patrizia.grifoni,tiziana.guzzo}@irpps.cnr.it

Abstract. The evolution of technologies is making available new devices that allow designing enriched interaction environments between humans and computers. New technological tools, such as multimodal platforms, can be used to enhance cognitive and learning processes. This paper introduces a new paradigm that includes different characteristics of the human-human communication, such as emotions, aesthetics, presence perception, and semantic communication that, usually, are not considered in the human computer communication. Moreover, it provides a gaming application scenario and the description of an Advanced Multimodal Platform for game-based Learning.

Keywords: Cognitive processes, Learning, Multimodal, Emotions, Aesthetics, Presence perception, Semantic communication.

1 Introduction

In Neurology, two different studies on rats demonstrated that the morphology and chemistry of the brain could be experientially altered. Therefore the belief that the brain was immutable and subject only to genetic control was proved wrong and the role of environmental influences in altering the structure of the brain in both animals and human beings was demonstrated. Since then, the capacity of the brain to respond to an enriched or impoverished environment has become an accepted fact among neuroscientists, educators and other professionals. In humans, as in many other species, brain development consists of continuous morphological reorganizations due to progressive and regressive changes including growth and degeneration of axons, dendrites and synapses as well as of entire neuronal circuits. Such a reorganization may occur at different times in different neuronal systems and different brain areas. At present, the specific function of single brain areas while performing high-level selective cognitive processes is still partially unknown. Similarly, neurophysiologic activation patterns are only schematically understood. Instead, it is clear that finding a solution to a given problem at the beginning of a learning process may require intense and deliberate effort; the demand of effort will diminish as skills improve and

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 562–570, 2011.
© Springer-Verlag Berlin Heidelberg 2011

increase. This well-known phenomenon can be observed in different domains and represents a central issue in the debate about the modalities of cognitive change and those learning processes that are accompanied by typical, possibly also topographical, activation changes. The evolution of technologies is making available new devices that allow designing enriched environment of interaction between humans and computers. Indeed, new technological tools such as multimodal platforms can be used in order to enhance cognitive and learning processes. For this reason, it can be useful to consider them as a part of a new communication and knowledge paradigm that involves humans and devices.

Starting from these experimental evidences and the technology evolution, the goals of the paper are:

- *the definition of a new paradigm that considers different characteristics of the human-human communication such as emotions, aesthetics, presence perception, semantic communication that, usually, are not considered in the human-computer communication;*
- *the introduction of a gaming application scenario and the description of an Advanced Multimodal Platform for game-based Learning.*

The new paradigm has to define how the environment, and then the interaction with the new ICT devices can stimulate and enhance cognitive and psychological aspects of human brain improving the brain activity and flexibility by means of the evocational point of view, where evocations are mind images or representations of the world (real and artificial).

The gaming application represents one particular type of edutainment applications that have the great potential to enhance cognitive and learning processes. In these environments, multimodal interfaces play a fundamental role for the achievement of a high degree of interactivity during the learning process. The platform will enable to define the interaction in a gamed-based learning environment through a multimodal interface. This platform allows efficiently managing multimodal communication between people participating in the virtual learning environment.

The paper is organized as follows. Section 2 describes the definition of a new paradigm that includes different characteristics of the human-human communication. In Section 3 a description of the gaming application scenario is given. Finally Section 4 concludes the paper.

2 The Definition of a New Paradigm

Cognitive and learning processes are partly influenced by teaching/learning approaches such as Collaborative and Cooperative learning, Discovery-based learning, Engaged learning, Problem-based learning, Whole-language approach and the like. These approaches have been largely investigated and offer a variety of procedures of implementation in different learning contexts and/or classrooms. The studies have considered subject and content of courses as well as a wide variety of social issues such as diversity, antisocial behavior, lack of prosocial values, egocentrism, alienation, loneliness, psychiatric pathologies, and much more. Even though such methods arguably have different influences on brain plasticity, to our best knowledge there are no significant studies that showed these impacts on the brain plasticity.

In recent years other aspects have been considered in learning/teaching processes such as the distinction between explicit (the "know what" type) and implicit (the "know how" type) knowledge that people use every day. People are usually conscious of explicit knowledge that is easy to communicate to others with confidence (I know that one plus one equals two); on the contrary implicit knowledge is hard to communicate to others because the know-how knowledge is largely hidden from our awareness (I know how to ski, but I cannot describe the exact actions necessary because many of them arise unconsciously). Another relevant aspect is the object perception and persistence in learning processes. Using different senses people acquire knowledge about discrete objects and events and maintain a representation of the sensorial perception of the objects over time, motion and features change. Sensorial experience consists of more than individual perceptions of objects of the world: we must also bind individual perceptions of objects into dynamic representations, which persist across time and motion, and perceive them as a coherent whole that leads to aesthetic and emotional responses. Without such a computation of persisting object hood, sensorial experience could be incoherent. In recent years, psychologists have made great strides in working out the principles that guide the construction and maintenance of representations of the same objects over time, motion and features change.

Modifications of the brain, explicit and implicit knowledge and object perception and persistence are important factors in teaching/learning processes and the implementation of technologies devoted to facilitate acquisition/sharing/transmission of knowledge and procedural skills should take into consideration how to improve these relevant factors. The use of emerging ICT technologies could define augmented environments and create "virtualized" spaces able to improve both brain structure and cognitive processes. In fact, the trend of the future ICT technologies is going to provide more human-like capabilities. The new paradigm has to consider different aspects of the human-human communication, such as emotion, aesthetics, presence perception, semantic communication that, usually, are not considered in the human computer communication.

Therefore the paper addresses the objectives specified in the following four different research topics: emotions, aesthetic or functional quality, the sense of presence and the semantic communication as described in the following sections.

2.1 Emotions

Emotions help us to guide our choices, avoid danger, and also play a key role in disambiguating non-verbal communication. Estimating user emotional states can greatly improve interaction quality by bringing it closer to human-human communication and by enabling automated feedback showing more empathy towards the user (affective computing). Affect estimation is proposed in a wide range of applications, such as identification of critical states during driving or dangerous operations, computer animation, rehabilitation including autism, learning and gaming, multimedia data tagging and management, marketing and advertisement [1] [2] [3].

Assessing emotions in real-life situations is thus essential to the understanding of human behavior as well as for designing affective computing systems. In this context, the research should be focus on the following issues:

- Can attention, interest and then emotions of a user be reliably estimated in a real life setting?
- How can contextual information be modeled, in a changing environment, and taken into account?
- How to integrate attention, interest, and emotions into a real environment such as gaming?

2.2 Aesthetic or Functional Quality

Aesthetic or functional quality is perceived by an individual and is at least partially subjective, and is closely linked to general emotional responses. Understanding aesthetic judgments is rarely investigated with scientific methods, because the absence of unbiased and robust experimental paradigms has limited progress.

Nevertheless, there is a recent surge of interest in this field from Neuroscience, supported by attempts to localize brain activity related to aesthetically pleasing stimuli [4]. Another promising avenue to make aesthetics accessible to scientific investigation is Interactive Evolutionary Computation (IEC), which has been applied to art and design problems where the aesthetic or functional quality that is perceived by an individual is at least partially subjective [5] [6]. Interactive Evolutionary Computation (IEC) has been applied to art and design problems to compute favorable solutions in large parameter spaces. IEC applications usually present a population of possible solutions from which the preferred constellations are selected by a human observer before the usual evolutionary operations are performed to produce the next generation; over many generations the population of solutions then approaches the most preferred ('optimal') constellation. Large population sizes and numbers of generations impose significant demands on the user. In this context, the research should be focus on the following issues:

- How to study aesthetic preferences objectively by selecting the 'best' composition by means of eye movements, which reduces user fatigue without sacrificing quality of fitness assessment?
- How to generalize these studies by defining and implementing robust and efficient methods and techniques to use attention for driving the development of exemplars for subjective evaluation is systematically?
- How behavioral experiments can be used to assess inherently subjective experiences of humans in their natural environment, and how this can be related to individual differences and overall group experiences?
- How individual and group responses to evolved aesthetics are reflected by brain activity?

2.3 The Sense of Presence

The sense of presence (i.e., a perceptual illusion of non-mediation) is a concept related to factors like high attention, interest, motivation, and involvement. The sense of presence might occur when an individual's perception is bounded to a mediated environment. Media are more and more shared by several users (i.e. collaborative learning platforms, social networks, etc …), leading to the inclusion of social experience in the presence concept. Presence can thus be divided in two components: spatial presence [7], the sensation of "being there" which corresponds to immersion, and social presence [8], the awareness of the others' presence. Several studies have investigated the role of media factors (content of the output, persistence, coherence across modalities) [8] and user factors (interest, involvement and attention) [7] in the elicitation of both spatial and social presence. Physiological signals can be used to infer the state of a person since they reflect the central and peripheral activity of the body. For instance, several studies have now demonstrated the existence of particular patterns of physiological activity for given emotions and emotional dimensions [9] [10] [11]. Attention is one of the key points necessary for the emergence of the presence feeling [7] and it also facilitates learning [12] [13].

The research questions are divided according to the two components of presence: spatial and social presence.

- How learning outcomes and spatial presence can be predicted on the basis of psycho physiological responses?

- Does the experience of spatial presence increase performance and learning?

- What is the role of social presence and interaction in facilitating performance and learning when playing digital games?

- Does psycho physiological synchronization across users (i.e., physiological compliance) contribute to stronger social presence, higher performance, and better (group) learning outcomes?

- How to improve social interaction and physiological compliance by adapting the mediated environment?

2.4 The Semantic Communication

The semantic communication is the capability for two intelligent beings to communicate meaningfully, without any common language or background. This capability is very relevant for developing new abilities in human-computer communication and computer-computer communication. In fact, the human-human communication is characterized by different "parameters" such as the content, its semantics and its aesthetical aspect, the emotional states and the attention of the actors involved in the communication, etc… A key role in efficient semantic communication is played by (i) the content management, (ii) the communication paradigm and (iii) the augmented environment framework. In order to provide intelligent management of the content, many evolutionary computation techniques, have been provided in the literature which seem better suited to be applied to dynamics environments and contents. The use of various evolutionary algorithms was proposed at multiple stages of the information retrieval process. Fan et. al. in [14] introduced genetic ranking function discovery framework. Nyongesa & Maleki-dizaji in [15] used interactive evolutionary learning for user modeling. On considering the communication paradigm, researches focused on the development of both perceptual systems similar to the human brain perceptual systems and models of the response of humans to the stimuli produced by the systems. Furthermore, considering the limits of human channels capacity, an open challenge is to maximize information exchange between people and computing systems in a way that is effortless for users. Finally, with respect to the augmented environment framework current trends in human computer interaction with virtual environments try to mimic the natural actions that are performed in the real world. As for instance, back in 2006, the commercial launch of the Wii, a video game console with inferior specifications than its competitors (namely Sony's PlayStation 3 and Microsoft's Xbox 360), but with the first mass market tracked wireless controller, resulted in an immediate success.

In this context, the research should be focus on the following issues:

- How can digital contents be meaningfully generated, indexed and clustered?

- How does the selection of meaningful contents have to be performed taking into account the user's intention, the history of the user interaction behavior, and the impact that these contents should have on the user's brain structure?

The paper requires the definition and development of a new human-system communication paradigm that augments "traditional" human-computer interaction to make it more effective in terms of human mental, cognitive, learning and interactive abilities. The new paradigm will stimulate the evolution of human perception ability and it will support the machine evolutionary production stimuli. This requires the identification of the features and parameters that characterize the human perception ability and, at the same time, it will identify the features and parameters characterizing the machine evolutionary generation of contents and visual stimuli.

The expected results of this investigation are twofold: a better understanding of how human visual perception operates and interacts with other senses and understanding of how next generation display technologies may be used for augmenting human perception, cognition, and learning. In this context, the research should be focus on the following issues:

- How can multimodal stimuli be processed in order to augment human perception, cognition, and learning?

- How can the impact of multimodal stimuli on human cognitive and learning abilities be assessed?

- How can the responses of the human brain and sensory system to the presentation of appropriate kinds of multimodal stimuli be modeled?

- How can several communication modalities be integrated for producing the most relevant contents considering affective reactions, aesthetic preferences and perception of the user?

- How can the system be continuously adapted considering the evolution of affective reactions, aesthetic preferences and perception of the user?

Finally, the paper requires the definition of an augmented environment framework that could be easily adapted at the final user level to different application scenarios defined by educators, teachers and physicians. This framework has to integrate the human – system communication concepts defined in other parts of the project related to the way the virtual objects have to be presented to the user, which behavior they have and which underlying rules exist in order to achieve the expected goal (in learning, in the care process etc...). Another key feature will be the interaction with the virtual environment, based on innovative devices and technologies that enable true multiuser cooperative environments. The augmented environment framework integrates several devices: OLED displays, multitouch surfaces, specialized audio, 3D tactile device, 3D-TOF cameras and eye-gaze tracking systems.

In this context, the research should be focus on the following issues:

- How can emerging technologies for collaborative and natural communication with the environment be integrated in an augmented environment framework?
- How does the augmented environment framework take into account complex issues such as emotions, aesthetics and sense of presence?

3 A Gaming Application Scenario

An important basic/applied research question is how cognitive and learning processes, predict different learning outcomes in the context of technology-mediated learning.

To answer this question, psycho physiological features (heart-rate variability, interparticipant synchrony, EEG energy) will be examined and alternative data analytic techniques will be compared to find the most useful ways to analyze the processes occurring during game playing. Game-based learning environments represent one particular type of edutainment applications that have the great potential to support learning contexts, as they provide a new form of engagement that is participatory and collaborative. The users interact with games under various conditions where the content, the form, and the modalities will be manipulated to affect presence occurrence. Learning outcomes will then be investigated with regards to attentional factors and the emergence of the presence state. In these environments, multimodal interfaces play a fundamental role for the achievement of a high degree of interactivity during the learning process. In Figure 1 an Advanced Multimodal Platform for game-based learning is provided, which enables the interaction in a gamed-based learning environment through a multimodal interface.

This platform allows efficiently managing multimodal communication between people participating in the virtual learning environment.

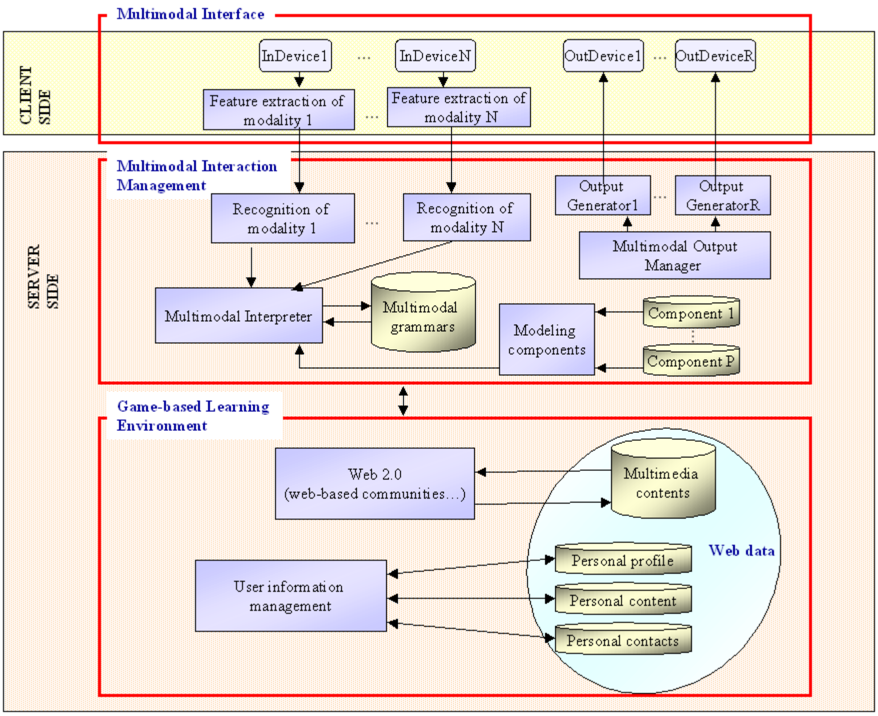


Fig. 1. Architecture of the multimodal environment

The platform is based on client-server architecture, each person (i.e. learner and teachers) can access to the platform from its own device that is equipped with a multimodal interface. Therefore, the platform client includes specific I/O devices, such as, for example, display, cameras, microphone, and loudspeakers, as well as the components for extracting features from the received signals. The feature extraction occurs on the client side, since it requires limited amount of memory and computational power, whilst the remaining recognition process, which consists in matching the extracted features with a predefined set of patterns, is completed on the server. The platform server consists of the multimodal interaction management and the game-based learning environment. The multimodal interaction management is responsible for recognizing unimodal input coming from the features extractors of each modality, appropriately interpreting these inputs, integrating these different interpretations into a joint semantic interpretation, and understanding which is the better way to react to the interpreted multimodal request by activating the most appropriate output devices. To do that, this component includes: the unimodal input recognizers; the multimodal interpreter that integrates the recognized inputs; the modeling components, that are aimed at capturing some information used during the interpretation phase for leading up to the most probable interpretation of the user input. The game-based learning environment consists of two main components: the Web 2.0 module, which provides social networking services, and the user information management, which is devoted to store and manage personal data of network members.

4 Conclusion

The paper provides the definition and development of a new human-system communication paradigm to augment "traditional" human-computer interaction and make it more effective in terms of human cognitive and learning activities. The new paradigm considers different characteristics of the human-human communication such as emotions, aesthetics, presence perception, semantic communication that, usually, are not considered in the human-computer communication. Moreover, in the paper an Advanced Multimodal Platform for game-based learning is provided which helps to enhance cognitive and learning processes as it makes the interaction with the game more easy, participative and less workload consuming than standard graphical interfaces.

References

1. Kasap, Z., Magnenat-Thalmann, N.: Intelligent virtual humans with autonomy and personality, vol. 1(1-2), pp. 3–15. IOS Press (2007)
2. Liu, C.K., Conn, N., Sarkar, W.S.: Online affect detection and robot behaviour adaptation for intervention of children with autism. IEEE Trans. Robotics 24(4), 883–896 (2008)
3. Chanel, G., Kierkels, J., Soleymani, M., Pun, T.: Short-term emotion assessment in a recall paradigm. Int. J. of Human-Computer Studies 67, 607–627 (2009)
4. Vessel, E.A., Starr, G.G., Rubin, N.: This is your brain on art Available at: [Abstract] 9(8):951, 951a (2009), http://journalofvision.org/9/8/951/, doi:10.1167/9.8.951
5. Bentley, P.J.: Generic Evolutionary Design of Solid Objects using a Genetic Algorithm. Huddersfield, University of Huddersfield. PhD (1996)
6. Funes, P., Pollack, J.: Computer Evolution of Buildable Objects. In: Bentley, P.J. (ed.) Evolutionary Design by Computers, pp. 387–404. Morgan Kaufmann, San Francisco (1999)
7. Wirth, W., Hartmann, T., Böcking, S., Vorderer, P., Klimmt, C., Schramm, H.: A process model of the formation of spatial presence experiences. Media Psychology 9, 493–525 (2007)
8. Lee, K.M., Nass, C.: Social-psychological origins of feelings of presence: Creating social presence with machine-generated voices. Media Psychology 7, 31–45 (2005)
9. Lang, P.J., Greenwald, M.K., Bradley, M.M., Hamm, A.O.: Looking at pictures: Affective, facial, visceral, and behavioral reactions. Psychophysiology 30(3), 261–273 (1993)
10. Lang, P.J., Bradley, M.M., Cuthbert, B.N.: International affective picture system (IAPS): Digitized photographs, instruction manual and affective ratings. University of Florida, Gainesville (2005)
11. Lisetti, C.L., Nasoz, F.: Using noninvasive wearable computers to recognize human emotions from physiological signals. Journal on Applied Signal Processing 11, 1672–1687 (2004)
12. Bryant, J., Fondren, W.: Psychological and communicological theories of learning and emotion underlying serious games. In: Ritterfeld, U., Cody, M., Vorderer, P. (eds.) The Social Science of Serious Games: Theories and Applications. Lawrence Erlbaum Associates, Mahwah (2009)
13. Graesser, A., Chipman, P., Leeming, F., Biedenbach, S.: Deep learning and emotion in serious games. In: Ritterfeld, U., Cody, M., Vorderer, P. (eds.) The Social Science of Serious Games: Theories and Applications. Lawrence Erlbaum Associates, Mahwah (2009)
14. Fan, W., Gordon, M.D., Pathak, P.: A generic ranking function discovery framework by genetic programming for information retrieval. Inf. Process. Manage 40(4), 587–602 (2004)
15. Nyongesa, H.O., Maleki-Dizaji, S.: User modelling using evolutionary interactive reinforcement learning. Inf. Retr. 9(3), 343–355 (2006)

A Decision Support System Based on Metaheuristic Model for Aircrafts Landing Problems

Sonda Hbaieb, Souhail Dhouib, and Habib Chabchoub

Research Unit of Logistic, Industrial Management and Quality,
Higher Institute of Industrial Management,
University of Sfax, Tunisia
souhail.dhouib@gmail.com

Abstract. This paper presents a Decision Support System (DSS) designed to aid the manager to optimize the scheduling aircrafts (planes) landings at an airport. This problem is one of deciding a landing time for each plane such that each plane lands within a predetermined time window and the landings of all successive planes are respected.

This DSS is developed using 4D language, it is composed of three components.

1) The Model Subsystem; the particle swarm optimization metaheuristic allows the optimization model for identification of aircraft that respect the preferences of the decision maker. 2) The Data Subsystem; the DSS uses a knowledge base composed of data publicly available from OR-LIBRARY, involving from 10 to 50 aircrafts (we consider 25 problems in total). 3) The Dialog Subsystem; a visual interactive simulation model will allow the interactive and convivial resolution.

Keywords: DSS, simulation, metaheuristic, Particle swarm optimization algorithm, transport, scheduling.

1 Introduction

The Aircraft Landing Problem (ALP) can be described as follows: a set of planes (*P*) with target landing times (*T*) and time windows ([*E,L*]) for landings are given, and runways (*R*) are given, the objective of the ALP is to minimize the total (weighted) deviation from the target landing time for each plane. This problem can be in static case (when the complete knowledge of the set of planes that are going to land is available), or in dynamic case (when the situation changes: the set of planes is modified every time a new aircraft arrives).

In [1] a mixed integer formulation of the static ALP is formulated and the problem is optimally solved with a linear programming based on tree search algorithm after relaxing binary variables and strengthening the formulation with additional constraints. On the other hand, heuristic approaches [10] are proposed for the ALP based on the segmentation of time. Also, [9] shows that a genetic algorithm and a branch and bound algorithm could be applied for the ALP. For the dynamic case of the ALP, a population heuristic method is used in [2]. As well, Pinol and Beasley in [18] develop a Scatter Search and Bionomic Algorithms to solve the ALP in the static

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 571–580, 2011.
© Springer-Verlag Berlin Heidelberg 2011

case. In a recent work, [7] generates a multi start adaptive variable neighborhood search metaheuristic for the same purpose. However, in literature the Particle Swarm Optimization (PSO) metaheuristic is not adapted for the ALP: this is our aim in this paper.

The remainder of the paper is organized as follows. In section 2, the formulation of the ALP is recalled. In section 3, the basic PSO metaheuristic is detailed. The values of PSO parameters are fixed in section 4. Computational results are reported and analyzed in section 5; the results of the proposed PSO metaheuristic are compared with those obtained by Scatter Search and Bionomic algorithms developed by [18]. Section 6 concludes the paper and suggests future research directions.

2 Problem Formulation

A plane gets information from air traffic control at an airport: an affected runway (R_i) and a fixed target landing time (T_i) belonging to a predetermined time window, bounded by an earliest landing time (E_i) and a latest landing time (L_i). This time window differs from one plane to another. The time required if a plane flies at its maximum airspeed corresponds to the earliest time. In addition, the time required if a plane flies at its most fuel efficient airspeed, while circling for the maximum allowable time, represents the latest target time (see [2]). Besides, each plane has a preferred speed (or most economical) namely the cruise speed. So, the fixed target landing time of a plane is the time required if this plane flies at its cruise speed.

Another parameter must be defined: the separation time between two planes. This separation time is a distance, depending on the type of the plane, which should be respected during landing (in order to maintain an aircraft's aerodynamic stability).

According to [1], the problem is to determine the landing time X_i and the allocation variable y_{ir} (if plane i lands on runway r) for each plane which gives the minimum cost.

In the static case of scheduling ALP, the computation is used for scheduling the optimal sequence for planes waiting to land at one or several runways. In this paper, we consider the case of the single runway applied on two different objectives (linear and non-linear objectives), both based on deviations from target times (see [18]).

2.1 Linear Objective

For each plane, costs of deviation are calculated to landing either earlier (g_i) or later (h_i) than a target landing time (T_i). The linear objective is to minimize the overall cost:

$$Min \sum_{i=1}^{P} (\alpha_i g_i + \beta_i h_i) \tag{1}$$

Where

$$\alpha_i = \max(0, T_i - x_i) \tag{2}$$

$$\beta_i = \max(0, x_i - T_i) \tag{3}$$

α_i is the deviation before target time for plane i, $i \in [1..P]$

β_i is the deviation after target time for plane i, $i \in [1..P]$

To satisfy the separation criteria between all pairs of plane, each plane must land on one of the runways within its predetermined time windows (see Fig. 1).

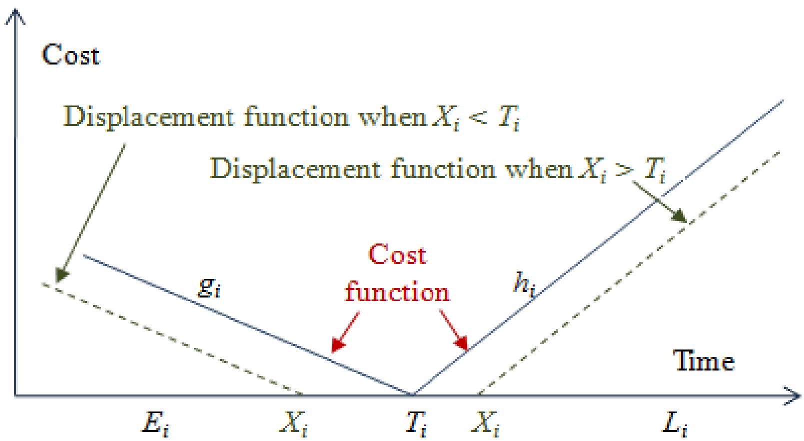


Fig. 1. Variation in cost for an aircraft within its time window

2.2 Non-linear Objective

As predefined in [18], the difference between the scheduled landing time and the target time (based on d_i) represents the non-linear function: where d_i is the deviation from the target time ($d_i = x_i - T_i$) for aircraft i, $i \in [1..P]$.

$$Min \sum_{i=1}^{P} D_i, \qquad D_i = \begin{cases} -d_i^2 & if\ \ d_i \geq 0, \\ +d_i^2 & otherwise \end{cases} \tag{4}$$

3 Basic Particle Swarm Optimization

The objective of a model subsystem is to model the decision environment. In this paper the PSO metaheuristic is used to provide sensitivity analysis support and thereby optimize the single runway ALP.

The PSO metaheuristic is introduced in 1995 by Kennedy and Eberhart (see [10]) it is based on a social-psychological model of social influence and social learning [12]. A PSO algorithm maintains a swarm of particles, it models the social behavior of bird or fish coaching, where each particle represents a potential solution (a swarm is similar to a population, while a particle is similar to an individual). The PSO metaheuristic has been successfully applied to many optimization problems. Firstly,

[11] uses PSO metaheuristic to optimize continuous nonlinear functions. Then, PSO metaheuristic is successfully applied to a wide range of applications such as permutation flow shop [13] and [14], vehicle routing problem [17], traveling salesman [5], vibration control of beams with piezoelectric sensors [15], data clustering [21] and chaotic maps [6].

The PSO metaheuristic is based on a memory of best individual previous and best neighborhood positions. These two best positions serve as the attractor to find optimal regions of high dimensional search space. For the first memory, the best personal (previous) position of each individual is the best position found by that individual. For the second memory, the best neighborhood position is the best position found by all the swarm since the start of the simulation.

Then, the position of each particle is adjusted according to its own experience and that of its neighbors. Let $x_i(t)$ denote the position of particle i in the search space at time step t. The position of the particle is changed by adding a velocity, $v_i(t)$, to the current position.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{5}$$

It is the velocity vector that drives the optimization process, and reflects both the experimental knowledge of the particle and socially exchanged information from the particle's neighborhood.

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \tag{6}$$

Where

- $v_{ij}(t)$ is the velocity of particle i in dimension $j=1..n$ at time step t,
- $x_{ij}(t)$ is the position of particle i in dimension $j=1..n$ at time step t,
- w is the inertia weight,
- c_1 and c_2 are positive acceleration constants used to scale the contribution of the cognitive and social components respectively, r_{1j} and r_{2j} are random values in the range [0,1].
- $y_i(t)$ is the personal best position associated with particle i,
- $\hat{y}(t)$ is the global best position at time step t.

This equation is composed of three terms: (1) the previous velocity, $v_i(t)$, which serves as a memory of the previous direction to disallow the particle from drastically modifying direction. (2) the cognitive component, $c_1 r_1 [y_i - x_i]$, which quantifies the performance of particle i relative to past performances. The effect of the cognitive component is that particles are drawn back to their own best positions, resembling the tendency of individuals to return to situations or places that most satisfied them in the past. (3) the social component, $c_2 r_2 [\hat{y} - x_i]$, which quantifies the performances of particle i relative to a group of particles. This social component ensures that each particle is also drawn towards the best position found by the particle's neighborhood.

Main steps of the algorithm are:

1. Initialize parameters
2. Initialize population
3. Repeat
4. Calculate fitness values of particles
5. Find the global best
6. Find the personal best
7. Calculate the velocities of particles
8. Update the particle positions
9. Until requirement is meat

4 PSO Components

An extensive study of different variants of the PSO metaheuristic is reported. The PSO is influenced by a number of control parameters: number of particles, acceleration coefficients, inertia weight and number of iterations.

4.1 Swarm Size

The swarm size indicates the number of particles in the swarm (*Ss*). A large swarm allows larger parts of the search space to be covered per iteration, compared to smaller swarms. It has been shown in a number of empirical studies that the PSO has the ability to find optimal solutions with small swarm sizes of 10 to 30 particles (see [4] and [23]).

In order to fix the size of swarm (*Ss*), we have tested the impacts of varying this parameter on the result of the objective function. An average relative gap is computed for the first instance (one runway and ten planes) to different sizes of the swarm (10, 12, 14, 16, 18, 20, 22, 24, 26, 28 and 30) with fixed computation time (1 second for each test). For example, to compute the first average relative gap for (*Ss* = 10): the developed PSO metaheuristic is replicated ten times, with fixed *Ss* size at value 10, in each iteration 1 second is allowed. Based on this test, the objective function development regarding the swarm size is depicted in Fig. 2. As shown in this figure, the average relative gap values are good at near *Ss* = 14.

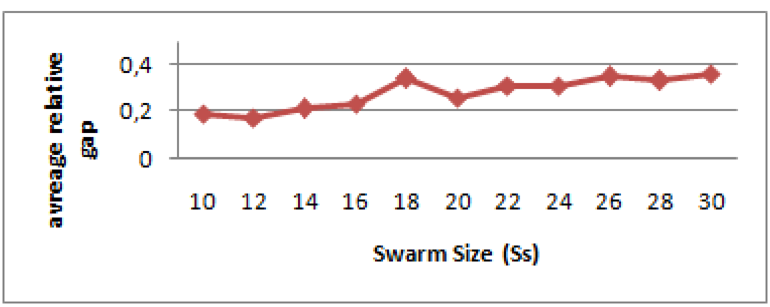


Fig. 2. Impact of the parameter swarm size (Ss) on the objective function

4.2 Inertia Weight

The inertia weight was introduced by [19] as a mechanism to control the exploration and exploitation abilities of the swarm. To vary the inertia weight, non-linear decreasing approaches are used, where an initially large value decreases nonlinearly to a small value. [24] and [25] prove that nonlinear decreasing methods allow a shorter exploration time than the linear decreasing methods, with more time spent on refining solutions. In the developed PSO metaheuristic the nonlinear decreasing method is used through of equation (7) described below, where $\alpha = 0.975$.

$$w(t+1) = \alpha w(t') \tag{7}$$

The inertia is only changed when there is no significant difference in the fitness of the swarm (the variation in particle fitness of a 20% subset of randomly selected particles). An initial inertia weight of $w(0)$=1.4 with a lower bound of $w(n)$=0.35 are fixed by [24] and [25]; the same parameters values are used for our developed PSO metaheuristic.

4.3 Acceleration Coefficients

The stochastic influence of the cognitive and social components on the overall velocity of a particle is controlled by the acceleration coefficients; c_1 and c_2, together with the random vectors r_1 and r_2 (see equation (6)). *c1* and *c2* are usually static, their optimized values are empirically found by [8]: $c1=c2=1.49618$. The same parameters values are used for our developed PSO metaheuristic.

4.4 Stopping Condition

A ratio is considered to define the stopping condition as defined by [22]. This ratio is based on the rate of change in the objective function: terminate when the objective function slope is approximately equal to zero.

$$f' = \frac{f(\hat{y}(t)) - f(\hat{y}(t-1))}{f(\hat{y}(t))} \tag{8}$$

If $f' = 0$ for a number of consecutive iterations, the swarm is assumed to have converged (determines if the swarm is still making progress using information about the search space).

5 Computational Results

Computational results for linear and non-linear objective functions are presented in this section, using instances publicly available from OR-Library, involving from 10 to 50 aircrafts (see Fig. 3). The PSO metaheuristic presented above is implemented in 4D language on a 2GHz Pentium PC with 2 GB of memory.

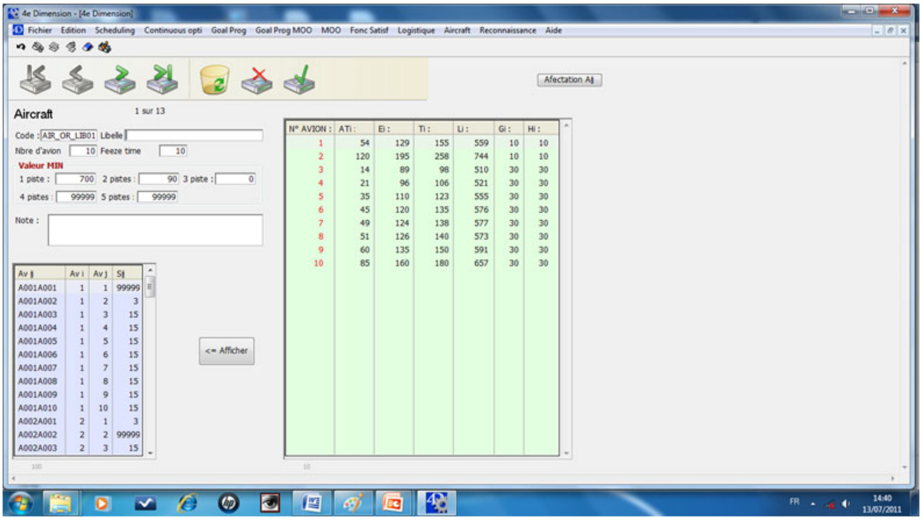


Fig. 3. The developed DSS

We have compared our method to the best known heuristics in the literature for the single runway ALP problem: Scatter Search algorithm (SS) and Bionomic Algorithm (BA) developed by [18]. Results for each algorithm and each objective are presented in Table 1 and Table 2 (Each heuristic was replicated ten times on each problem; all problems instances are publicly available from OR-Library). The notation used in this table is: *P* number of planes, Z_{opt} value of the optimal or best known solution, *T* total execution time in seconds for the ten replications and G_b percentage gap associated to the best solution found over the ten replications.

Table 1. Results for linear objective

P	Z_{opt}	*SS*		*BA*		*PSO*	
		T	G_b	*T*	G_b	*T*	G_b
10	700	4	0	60	0	67	0
15	1480	6	0	90	0	58	0
20	820	8	0	99	0	102	0
20	2520	8	0	95	0	86	0
20	3100	9	0	100	0	136	0
30	24,442	158	0	274	0	318	0
44	1550	195	0	79	0	167	0
50	1950	42	52.05	287	36.15	253	38.64
	Average	**53.75**	**6.50**	**135.50**	**4.52**	**148,37**	**4.83**

In the linear context, the equation (1) is used to evaluate the PSO metaheuristic. As it can be seen in table 1, the average deviations from the best known solutions for the SS and the BA algorithms, developed by [18], are respectively (6.50%) and (4.52 %). Whereas for the proposed PSO metaheuristic, the average deviation is (4.83 %). For the BA algorithm, both average deviation and computational time are the best. Besides, the PSO average deviation results are better than SS algorithm while the PSO computational time is greater than SS one.

Table 2. Results for non-linear objective

P	***Z_{opt}***	***SS***		***BA***		***PSO***	
		T	***G_b***	***T***	***G_b***	***T***	***G_b***
10	4849	6	0	57	0	52	0
15	18,337	8	0	56	0	63	0
20	35,632	9	0.28	70	0.28	94	0.28
20	20,001	9	0.95	70	0.59	85	0.62
20	19,381	9	6.11	71	1.03	103	1.57
30	-2,847013	15	0	68	0	58	0
44	-23,266	24	0	89	0	87	0
50	728,837	23	19.33	122	4.65	146	5.62
	Average	**12.88**	**3.33**	**75.37**	**0.82**	**86**	**1.01**

In the non-linear context, the equation (4) is used to evaluate the PSO metaheuristic. Table 2 shows that the best average deviation is given by BA (0.82 %). Concerning the proposed PSO, it gives an average deviation less than this given by SS, nevertheless it takes the most important computational time.

6 Conclusions and Future Work

In this paper a DSS is presented to optimize the single runway Aircraft Landing Problem (ALP) using Particle Swarm Optimization (PSO) metaheuristic. This DSS is purposed to provide support and not replace effective decision making: the PSO metaheuristic is allowed to the Model Subsystem; the knowledge base composed of data publicly available from OR-LIBRARY is enriched the Data Subsystem and the 4D language is used to develop the Dialog Subsystem.

Further research directions would be to hybrid the proposed PSO metaheuristic with local search methods, such as variable neighborhood search metaheuristic, which leads to better results but at the price of very high computational requirements.

References

[1] Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D.: Scheduling aircraft landings-the static case. Transportation Science 34, 180–197 (2000)

[2] Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D.: Displacement problem and dynamically scheduling aircraft landings. Journal of the Operational Research Society 55, 54–64 (2004)

[3] Beasley, J.E., Sonander, J., Havelock, P.: Scheduling aircraft landings at London Heathrow using a population heuristic. Journal of the Operational Research Society 52, 483–493 (2001)

[4] Brits, R., Engelbrecht, A.P., Van den Bergh, F.: A Niching Particle Swarm Optimizer. In: Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning, pp. 692–696 (2002)

[5] Chen, S.M., Chien, C.Y.: Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Systems with Applications. Uncorrected Proof, Available online May 1 (in press, 2011)

[6] Chuang, L.Y., Yang, C.H., Li, J.C.: Chaotic maps based on binary particle swarm optimization for feature selection. Applied Soft Computing 11(1), 239–248 (2011)

[7] Dhouib, S.: A Multi Start Adaptive Variable Neighborhood Search Metaheuristic for the Aircraft Landing Problem. In: Proceedings of the IEEE-Explore International Conference on Logistics, pp. 197–200 (July 2011)

[8] Eberhart, R.C., Shi, Y.: Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, pp. 84–88 (July 2000)

[9] Ernst, A.T., Krishnamoorthy, M., Storer, R.H.: Heuristic and exact algorithms for scheduling aircraft landings. Networks 34, 229–241 (1999)

[10] Jung, G., Laguna, M.: Time segmenting heuristic for an aircraft landing problem. Working paper, Leeds School of Business, University of Colorado, Boulder, CO Currently (2003), `http://leeds.colorado.edu/Faculty/Laguna/articles/tsh.html`

[11] Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1942–1948 (1995)

[12] Kennedy, J., Mendes, R.: Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. In: Proceedings of the IEEE International Workshop on Soft Computing in Industrial Applications, pp. 45–50 (June 2003)

[13] Lian, Z., Gu, X., Jiao, B.: A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan. Applied Mathematics and Computation 175(1), 773–785 (2006)

[14] Liao, C.J., Tseng, C.T., Luarn, P.: A discrete version of particle swarm optimization for flowshop scheduling problems. Computers & Operations Research 34(10), 3099–3111 (2007)

[15] Marinaki, M., Marinakis, Y., Stavroulakis, G.E.: Vibration control of beams with piezoelectric sensors and actuators using particle swarm optimization. Expert Systems with Applications 38(6), 6872–6883 (2011)

[16] Marinakis, Y., Marinaki, M., Dounias, G.: A hybrid particle swarm optimization algorithm for the vehicle routing problem. Engineering Applications of Artificial Intelligence 23(4), 463–472 (2010)

[17] Nickabadi, A., Ebadzadeh, M.M., Safabakhsh, R.: A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing 11(4), 3658–3670 (2011)
[18] Pinol, H., Beasley, E.: Scatter Search and Bionomic algorithms for the aircraft landing problem. European Journal of Operational Research 171, 439–462 (2006)
[19] Shi, Y., Eberhart, R.C.: A Modified Particle Swarm Optimizer. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 69–73 (May 1998)
[20] Soomer, M.J., Franx, G.J.: Scheduling aircraft landings using airlines' preferences. European Journal of Operational Research 190, 277–291 (2008)
[21] Tsai, C.Y., Kao, I.W.: Particle swarm optimization with selective particle regeneration for data clustering. Expert Systems with Applications 38(6), 6565–6576 (2011)
[22] Van den Bergh, F.: Particle Swarm Weight Initialization in Multi-Layer Perceptron Artificial Neural Networks. In: Proceedings of the International Conference on Artificial Intelligence, pp. 42–45 (1999)
[23] Van den Bergh, F., Engelbrecht, A.P.: Effects of Swarm Size on Cooperative Particle Swarm Optimizers. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 892–899 (2001)
[24] Venter, G., Sobieszczanski-Sobieski, J.: Multidisciplinary Optimization of a Transport Aircraft Wing using Particle Swarm Optimization. In: Ninth AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization (2002)
[25] Venter, G., Sobieszczanski-Sobieski, J.: Particle Swarm Optimization. Journal for AIAA 41(8), 1583–1589 (2003)

Comparing XML Files with a DOGMA Ontology to Generate Ω-RIDL Annotations

Nadejda Alkhaldi and Christophe Debruyne

Semantics Technology and Applications Research Lab,
Vrije Universiteit Brussel, Brussels, Belgium
{nadejda.alkhaldi,christophe.debruyne}@vub.ac.be

Abstract. To facilitate the process of annotating data in the DOGMA ontology-engineering framework, we present a method and tool for semi-automatic annotation of XML data using an ontology. XML elements are compared against concepts and their interrelations in the ontology using various metrics at different levels (lexical level, semantic level, structural level, etc.). The result of these metrics are then used to propose the user a series of annotations from XML elements to concepts in the ontology, which are then validated by that user. Those annotations - expressed in Ω-RIDL - are then used to transform data from one format into another format. In this paper, we demonstrate our approach on XML data containing vendor offers in the tourism domain, more precisely holiday packages.

Keywords: Annotation, Ontology/XML Matching.

1 Introduction

In the Community Driven Request for Proposals (COMDRIVE) [4] project, a platform was developed in which a potential customer's purchase intent is matched against existing vendor offers. Those offers - described in XML - came from autonomously developed information systems and were therefore different in structure and the kind of content they provided. In order for such a platform to be successful, a common from understanding of the domain from both the vendor and customer's side needs to be established and captured in an ontology, a [formal,] explicit specification of a [shared] conceptualization [7].

In the DOGMA [13] ontology engineering framework a special controlled natural language called Ω-RIDL [16,15] - both which will be explained later on in Section 2 - was developed to annotate data sources with concepts and relations from the ontology. Those annotations can then be used to, for instance, transform data from one format into another format or consult the data through the ontology. In the COMDRIVE RFP project, annotation was done manually. However, this process requires the person annotating the data to examine the structure and content of the data and compare it to the ontology. In this paper, we present a method and tool, presented in Sections 3, to facilitate the annotation process by having an agent examine the data and suggest the user annotations. The user

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 581–590, 2011.
© Springer-Verlag Berlin Heidelberg 2011

will thus have a hint to what the meaning of his data *could* be, after which he can decide to keep, edit or extend the generated annotation. The tool will thus be able to aid the user in *deciding* on a particular annotation (or commitment to) to the ontology. The annotations will be the result of *matching* the structure and content of the XML files against the concepts and relations in the ontology. We demonstrate our method and tool using the data used by the project in Section 4 before concluding and presenting our future directions in Section 5.

2 Background

In this section we first describe the DOGMA ontology engineering framework and commitments described in Ω-RIDL. We then go over to a brief state-of-the-art on various matching techniques relevant for this paper.

2.1 Developing Ontology Guided Methods and Applications

DOGMA [13] is an ontology approach having some characteristics that make it different from traditional ontology approaches such as its groundings in the linguistic representations of knowledge and the methodological separation of the domain- and application-conceptualization [12]. The knowledge building blocks - called *lexons* [9] - only need to express "plausible" facts (as perceived by the community of stakeholders) in order to be entered into the *Lexon Base*, a repository containing large sets of such lexons. A lexon is formally described as a 5-tuple $(\gamma, t_1, r_1, r_2, t_2)$, where γ is an abstract *context identifier* pointing to a resource such as a document on the Web or the community it originated from. The term t_1 plays the role of r_1 on term t_2 and t_2 plays the role of r_2 on t_1. The context identifier is assumed to identify unambiguously (to human users at least) the concepts denoted by the term and role labels.

The *Commitment Layer* contains ontological commitments made by users that use a selection of lexons to annotate data sources and specify constraints defining the use of the concepts in the ontology. DOGMA distinguishes two types of ontological commitments: *community commitments* and *application commitments*. The first denotes a meaningful selection of lexons and constraints that capture well the intended semantics of a domain. A community commitment corresponds with the ontology shared across stakeholders. In our approach, we use the ontology developed with a collaborative ontology engineering approach built around DOGMA, called DOGMA-MESS [3]. Details on the construction of this ontology have been reported elsewhere [4]. The latter extends the community commitments with mappings describing how one individual application commits to the ontology and are made with a special controlled natural language that we will explain in the following Section.

DOGMA is a fact-oriented modeling approach that is communication oriented, which means that the community comes to a shared understanding of the domain by communicating facts (lexons) rather then concepts, attributes and relations.

This differs from other frame-oriented formalisms such as OWL in which one can choose to model whether a particular relation in the world becomes an attribute or entity (denoting the relation) at design time. When a DOGMA ontology is transformed into OWL, these decisions are based on the constraints on those facts. This reduced to complexity of creating ontologies for the users.

2.2 *Ω*-RIDL

Ω-RIDL [16,15] allows users to describe how instances of concepts and their interrelations inside an information system commit to an ontology by: (i) selecting lexons from the Lexon Base, (ii) constraining over those lexons to capture well the intended meaning of that system and (iii) providing mappings between application symbols (e.g., XPaths[1] or fields in a table) and terms in the selected lexons. Examples of such constraints are frequency and totality constraints. The taxonomic relation can also be specified in a separate section (the default relation for taxonomy is "is a/subsumes"). It allows subtypes of concepts to inherit the relations of their supertype. *Ω*-RIDL provides a controlled natural language by allowing constraints and mappings to be expressed in terms of the natural language sentences created with lexons. An example of such a commitment can be found in Fig. 1 below. The application symbols mapped in this example are XPaths expressions and come from one of the XML files provided by vendors.

```
BEGIN SELECTION
``Product Community'' Product with / of Name
``Product Community'' Product with / of Description
``Tour Operator Community'' Holiday Package is a / subsumes Product
...
END SELECTION
BEGIN CONSTRAINTS
Holiday Package is identified by Name.
Holiday Package has exactly 1 Name.
Holiday Package has at most 1 Description.
...
END CONSTRAINTS
BEGIN MAPPING
map ``/items/item'' on Holiday Package.
map ``/items/item/title'' on Name of Holiday Package.
map ``/items/item/description'' on Description of Holiday Package.
...
END MAPPINGS
```

Fig. 1. Example of a commitment for a particular application showing pieces of the three parts: selection, constraints and annotations (or application symbol mappings)

[1] http://www.w3.org/TR/xpath/

2.3 Related Work on Matching Techniques

In this section we will present a brief survey on schema and ontology matching techniques. We have looked at several existing techniques for schema matching [8,19], database schema matching [5], matching XML schema to OWL [1], aligning and matching OWL ontologies [14,2], a method that detects similarities between two versions of an XML file [17], and a general framework for computing semantic similarity [11]. Some of those techniques use a linguistic match [8,19,2,11] using a thesaurus [8,2], a lightweight domain ontology [19]. [11] uses WordNet [6]. [8,19,1,14,2,17] also take into account the structures of the documents to be compared. Machine learning techniques are used by [19,5,14]. [8,5] also compare the data types and [19,5,14] even look at the similarity of values. [8,1,17] look at the keys (and referential constraints) of the XML schemas, [5] looks at the keys and integrity constraints in the base. [1] is discovering semantic isomorphic tree in an ontology based on the structure and also mentioned to look at the cardinality constraints of the schema and ontology.

3 Method

In our particular case, we start from an XML file without an XML schema and a DOGMA ontology. Since we have no XML schema, we cannot rely on the constraints, structure and data types typically provided by such a schema. Because of the above-mentioned strict separation of schema and instances in the DOGMA Ontology Engineering framework, we can only map elements of the XML file to facts in the ontology. Moreover, the goal is to annotate the instances in the XML *with* the DOGMA ontology. Based on these constraints, we decide to use string matching techniques to overcome spelling errors or slight variations in words (e.g., UK English vs. US English, concatenation of words, etc.), linguistic matching to look at the linguistic features of concepts in both schemas (e.g. to discover synonyms) and a structural match to detect the relation between two nodes that are not necessarily directly related to each other. We will now briefly describe how these matches can be used to generate Ω-RIDL mappings. We furthermore need a way to describe the type of a XML element based on the contents of one its attributes or contents of tags. For instance, `<facility type=''bar''>` should be mapped onto Bar, but not onto `Facility`, of which `Bar` is a subtype.

Fig. 2 gives the general idea of our Ω-RIDL mapping generator. It takes as input the lexons of the DOGMA ontology and an XML file. The XML file *must be related* to the ontology's application domain in order to obtain meaningful results. The user also defines a series of weights and thresholds to select to most successful mappings. The external resources are thesauri, reference ontologies or WordNet. In essence, the framework is extensible enough to plug in additional matching strategies. In this paper, we only took into account WordNet.

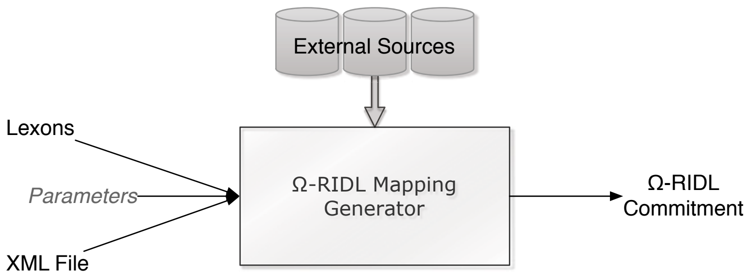


Fig. 2. The method. Users give an XML file and a DOGMA ontology as well as a series thresholds and weights for the particular heuristics.

3.1 Heuristics

In this section, we describe the different heuristics applied by Ω-RIDL mapping generator.

1. **Element Match.** This heuristic gives a similarity score based on a string metric between two labels, one of the XML (tag or attribute) and a concept in the ontology. We chose Jaro-Winkler [18] for its tolerance for spelling mistakes and variations over different dialects. As the interval of the scores lies between 0 and 1, the scores need not to be normalized.
2. **Linguistic Match** uses WordNet to give similarity score for XML and ontology concepts' names exploiting their linguistic features. This heuristic returns a score of 1.0 if both names are synonyms. 0.8 is returned if name of an ontology concept is a hyponym (narrower term) or a hypernym (broader term) of XML concept's name. Those scores are based on [2].
3. **Content Match** looks at the text inside XML tags and attributes (the previous two heuristics look at the label of XML tags and attributes) and matches those to the terms in the ontology using a string match.

When all those matches are applied, Ω-RIDL mapping generator computes their weighted mean based on a score that every match gives and its assigned weight. After that the fourth heuristic, structural match, is applied.

4. **Structural Match** adjusts the previously computed weighted means by looking to the structure of both the ontology graph and XML-tree. If two nodes from the XML and ontology graph are similar and have similar parents, then those nodes obtain a score of 1.0 for this heuristic. If the nodes are similar and do not have similar ancestors, we go up in the XML-tree's hierarchy and we look at the subtypes of the parent in the ontology graph. The ancestor of the XML node is compared to the subtypes and if a match between those two is found, the taxonomic relation is kept as a reference. A maximum number of steps over the XML-tree is specified by the user, and with every step the similarity score decreases by 0.1. If a match were not directly found, but after two steps, for example, the score would be 0.8. Fig. 3 depicts this process graphically.

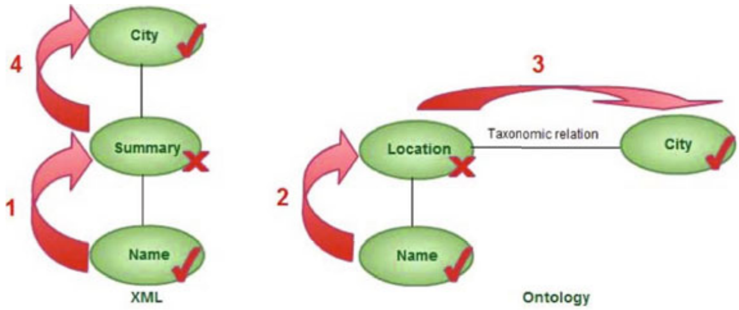


Fig. 3. The structural match traverses both graphs upwards to find common ancestors. The XML contained the name of a city inside a summary tag, whereas locations have names and city is a subtype of location in the ontology. After retrieving the parents of "Name" in both graphs (1, 2), we see they do not correspond. We then look for the parent of "Summary" in the XML graph (3) and compare it to the subtypes of "Location" in the ontology graph.

To summarize: using an XML and a DOGMA ontology, a series of mapping scores are calculated based on element, linguistic and content match. Those scores are then refined using the structural match. The refined scores are then compared against a threshold to produce the Ω-RIDL mappings.

4 Experiment

We tested our method and tool using the data obtained by the COMDRIVE RFP project. The data contains information about holiday packages in the winter sports domain. Those ontologies were developed in a modular way: a "general purpose" product ontology was extended with concepts suited for (winter sports) holiday packages [4]. Fig. 4 show a couple of such lexons. The structure of the XML file we are going to use is shown in Fig. 5. In this figure, the leaf nodes represent the text nodes in the XML.

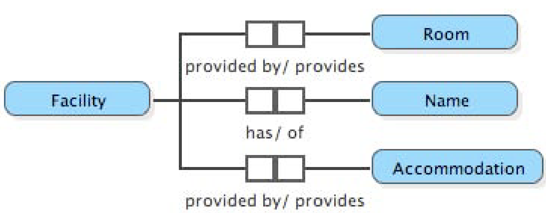


Fig. 4. Some lexons describing domain knowledge in the tourism domain. These lexons describe that both rooms and accommodations can have facilities, which in turn have a name.

When a user runs the Ω-RIDL mapping generator, the tool asks the user to provide a DOGMA ontology and XML file. The user then selects the heuristics to be used and assigns a weight to every heuristic. The user furthermore chooses

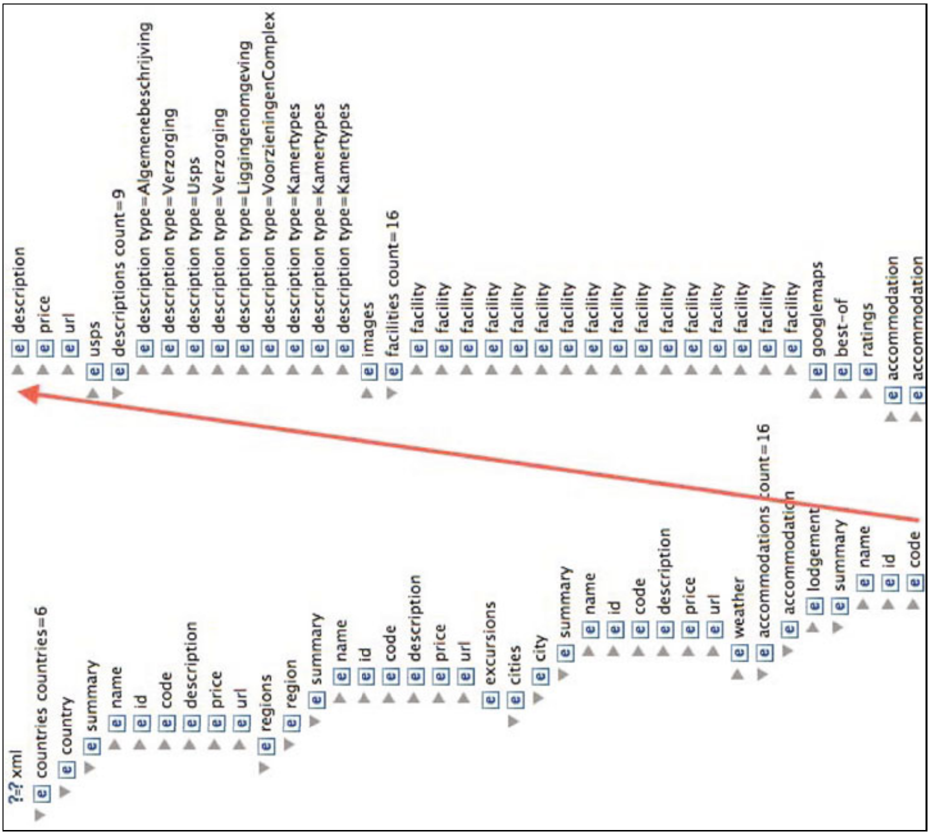


Fig. 5. Structure of the XML file used in our demonstration. Leaf nodes in this tree represent text nodes in the XML.

a depth for the structural match, and specifies thresh. The system takes those parameters and starts the mapping process. A series of mappings is then generated and used to output a selection and mappings (as described in section 2.2). The user can then use the generated mappings to *get an idea* how his application can commit to the ontology. In the example below, you can see that are two proposals for mapping the “region” tag: one to the ski area (destination of a holiday package) and on to the region. In the ontology, ski area is a subtype of region. As the agent cannot decide, the user should choose the most appropriate mapping (the last one, as it contains the ski area of all accommodations of all cities in that region). The result of content matching is a predicate to select elements in an XML that fulfill a certain condition, e.g., tags containing the word “Bar” can denote instances of Bar (a subtype of facility).

```
map ‘‘/countries/country/sumary/code’’ on
    Code identifies / identified by Commodity.
```

```
map ''/countries/country/regions/region'' on Region.
map ''/countries/country/regions/region'' on
    Ski Area destination of / with destination Holiday Package.
map ''/countries/country/regions/region/cities/city'' City.
map ''/countries/country/regions/region/cities/city/accommodations/
    accommodation/facilities/facility[text()='Bar']'' on Bar.
map ''/countries/country/regions/region/cities/city/accommodations/
    accommodation/facilities/facility[text()='Sauna']'' on Sauna.
map ''/countries/country/regions/region/cities/city/accommodations/
    accommodation/googlemaps/latitude'' on Latitude of / with Location.
map ''/countries/country/regions/region/cities/city/accommodations/
    accommodation/googlemaps/longitude'' on Longitude of/with Location.
```

When a user chooses an inappropriate combination of weights and threshold, however, the system proposes quite a few mappings that are making no sense. Examples of such mappings are shown below. Future work will consist of looking at appropriate combinations of weights and threshold for which more experiments and data will be needed. In the following mappings, we see how a description of a region is wrongly accepted as a description of an RFP and the name of a city to the name of a facility. The third mapping, however, is more interesting. The ontology contains the fact that commodities have prices. The price of a region in the XML however, contains the lowest price for accommodation in that region. The structural match thus also needs to be revisited to cope with such cases.

```
map ''/countries/country/regions/region/summary/description'' on
    Description of / has RFP.
map ''/countries/country/regions/region/cities/city/name'' on
    Name of/has Facility.
map ''/countries/country/regions/region/summary/price'' on
    Price of / has Commodity.
```

In this experiment, the four heuristics were able to tackle the problems described in Section 3 (variation is spelling, synonyms, identifying the type based on the textual content and choosing an appropriate type based on the structure). The weights and threshold, however, needed to be appropriately chosen in order to come up with good results. For this dataset, using 0.5 or 0.6 for the overall threshold value seemed to do well. If the threshold is higher, some important mappings are lost (especially the ones constructed by the content match because they most likely get a score of 0.0 from the other matches). For that reason we also advise to assign the content match - if used - quite an important weight. Structural match does not have a weight because it is used to adjust the final score given to every candidate match. There is also a special threshold for the element match. We recommend using at least 0.7 because element match can give a relatively high score for words that are very different (e.g., 0.55 when comparing "Accommodation" with "Military airport" and 0.6 for "Accommodation" with "Date and Time)". But we did not notice any case where it gave 0.7 or more for irrelevant words.

5 Conclusions

In this paper, we presented a method and tool for semi-automatically annotating XML XPaths to concepts in a DOGMA ontology using Ω-RIDL. For this, we defined four heuristics, each with a different purpose. Element match compares labels of XML tags and attributes against labels of ontology concepts. Linguistic match can map words linguistically related to each other (synonyms, hypernym, hyponym, etc.). Content match looks to the text of the tags and attributes in XML and helps to produce more accurate mappings. Structural match looks at the surrounding concepts or a particular concept (n deep) to determine the correct relation in the ontology to be mapped with.

Even though the tool generates quite a few false positives, which depends on the weights and thresholds chosen, the tool does give the user an idea about the kind of data in the XML file. The tool can thus be used to guide the user in the manual process of annotating the XML file's structure by means of, for example, an autocompletion user interface element in the editor.

Future work is threefold. Firstly, we need extra experiments to determine an appropriate combination of weights and thresholds. We furthermore only took into account the mappings of a commitment. Secondly, even though we do not rely on an XML schema from which some constraints such as mandatoriness can be deduced, we can look at the nodes and the contents of those nodes to infer similar constraints. If a particular tag has exactly one other tag every time it occurs, a mandatory and uniqueness constraint can be proposed to the user.

And thirdly, we mainly focused at the labels of both XML and ontology concepts. Whenever two relations between two concepts occurs in the ontology and two XML concepts match relatively well to those concepts, both relations are proposed and it's up to the user to decide what to keep. Sometimes, however, the labels of the XML tags contain enough information to choose the appropriate relation, e.g., the XML tag `<startDate>` can be matched with the concept `Date` playing the role `start of` on another concept in the DOGMA ontology.

References

1. An, Y., Borgida, A., Mylopoulos, J.: Constructing complex semantic mappings between xml data and ontologies. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 6–20. Springer, Heidelberg (2005)
2. Castano, S., Ferrara, A., Montanelli, S.: Matching Ontologies in Open Networked Systems: Techniques and Applications. In: Spaccapietra, S., Atzeni, P., Chu, W.W., Catarci, T., Sycara, K. (eds.) Journal on Data Semantics V. LNCS, vol. 3870, pp. 25–63. Springer, Heidelberg (2006)
3. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A Meaning Evolution Support System for Interorganizational Ontology Engineering. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 189–202. Springer, Heidelberg (2006)

4. Debruyne, C., Meersman, D., Baert, M., Hansenne, R.: Community driven request for proposals: Applying semantics to match customer purchase intents to vendor offers. In: Proc. of 7th International Conference on Web Information Systems and Technologies (WEBIST 2011). SciTePress (2011)
5. Dhamankar, R., Lee, Y., Doan, A., Halevy, A.Y., Domingos, P.: iMAP: Discovering complex mappings between database schemas. In: Weikum, G., König, A.C., Deßloch, S. (eds.) SIGMOD Conference, pp. 383–394. ACM, New York (2004)
6. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
7. Gruber, T.R.: Toward principles of the design of ontologies used for knowledge sharing. International Journal of Human and Computer Studies 43, 907–928 (1995)
8. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: Apers, P.M.G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., Snodgrass, R.T. (eds.) VLDB, pp. 49–58. Morgan Kaufmann (2001)
9. Meersman, R.: Semantic Ontology Tools in IS Design. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS, vol. 1609, pp. 30–45. Springer, Heidelberg (1999)
10. Meersman, R., Dillon, T., Herrero, P. (eds.): OTM 2010. LNCS, vol. 6427, pp. 727–744. Springer, Heidelberg (2010)
11. Pirrò, G., Euzenat, J.: A semantic similarity framework exploiting multiple parts-of speech. In: [10], pp. 1118–1125
12. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. SIGMOD Record Special Issue 31(4), 12–17 (2002)
13. Spyns, P., Tang, Y., Meersman, R.: An ontology engineering methodology for DOGMA. Applied Ontology 3(1-2), 13–39 (2008)
14. Straccia, U., Troncy, R.: oMAP: Combining Classifiers for Aligning Automatically OWL Ontologies. In: Ngu, A.H.H., Kitsuregawa, M., Neuhold, E.J., Chung, J.-Y., Sheng, Q.Z. (eds.) WISE 2005. LNCS, vol. 3806, pp. 133–147. Springer, Heidelberg (2005)
15. Trog, D., Tang, Y., Meersman, R.: Towards Ontological Commitments with Ω-RIDL Markup Language. In: Paschke, A., Biletskiy, Y. (eds.) RuleML 2007. LNCS, vol. 4824, pp. 92–106. Springer, Heidelberg (2007)
16. Verheyden, P., De Bo, J., Meersman, R.: Semantically Unlocking Database Content through Ontology-based Mediation. In: Bussler, C.J., Tannen, V., Fundulaki, I. (eds.) SWDB 2004. LNCS, vol. 3372, pp. 109–126. Springer, Heidelberg (2005)
17. Viyanon, W., Madria, S.K.: Xml-sim-change: Structure and content semantic similarity detection among xml document versions. In: [10], pp. 1061–1078
18. Winkler, W.E.: The state of record linkage and cur- rent research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04 (1999)
19. Xu, L., Embley, D.W.: Discovering direct and indirect matches for schema elements. In: Database Systems for Advanced Applications, pp. 39–46. IEEE Computer Society (2003)

On the Relation between Decision Structures, Tables and Processes

Willem De Roover and Jan Vanthienen

Department of Decision Sciences & Information Management,
Katholieke Universiteit Leuven, Belgium
{willem.deroover,jan.vanthienen}@econ.kuleuven.be

Abstract. Complex decisions are often modeled as processes that act as a guide for decision making. This however result in a serious lack of tracability, maintainability and difficulties during the (re)design of the decision. To solve this issue we propose an approach to start from real known decision structures (like decision table hierarchies) and transform these into process models. The modeling of a decision happens by means of a decision-goal tree which is closely related to the structure of decision tables. This decision-goal tree is then transformed into several process models which are evaluated and compared to each other by some criteria.

1 Introduction

Most business processes and business process models include decisions of some kind. For example, in Business Process Management and Notation (BPMN) [1], decisions are made in flow forks or are represented by a diamond. Sometimes, if the decision is more complex, the entire decision can be included as an activity or as a service (a decision service). Typical decisions are: creditworthiness of the customer in a financial process, claim acceptance in an insurance process, eligibility decision in social security, etc. The process then handles a number of steps, shows the appropriate decision points and represents the path to follow for each of the alternatives.

Moreover, in a large number of cases, the process does not just contain decisions, but the entire process is about making a decision. The major purpose of a loan process e.g., or an insurance claim process, etc., is to prepare and make the final decision. The process shows different steps, models the communication between parties, records the decision and returns the result.

It is, however, strange to observe how these business process models are describing (and automating) the process about major decisions, without first modeling the decision itself. Decisions are based on decision criteria [2], require one or more subdecisions [3], use a simple or complex decision technique [4] , conclude one or more results, etc. The decision process, however, is not the same thing as the decision structure [5].

In this paper we start from the structure of the decision such that the decision process can be organized according to various business specific critera. The

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 591–598, 2011.
© Springer-Verlag Berlin Heidelberg 2011

modeling of a decision happens with the help of a decision-goal tree, which is basically equal to a decision table hierarchy. Once the structure of a decision has been modeled the next step is to translate the decision into various process models. However the choice between various process models forms an important issue, which is why we propose a set of criteria to rate and compare the process models to each other.

This paper is structured as follows. Section 2 gives a summary of the relevant literature. In section 3 we discuss the decision table structures and introduce a decision-goal tree, a new concept for the modeling of decision. Section 4 reports how the decision-goal tree can be translated into decision process models that can be evaluated and compared to each other. We conclude in section 5.

2 Related Work

Browne et al. [6,7] define a structure in which high level goals are decomposed into goal graphs that are finally used to design a process model. However Browne only derives one execution path where we would like to derive many and compare them to each other. Vanderfeesten [8] chooses an approach in which a concept named Product Based Workflow Design is defined. The product is defined by means of a Product Data Model (PDM) which shows similarities with the Bill-Of-Material Concept.We distinguish ourselves from Vanderfeesten's approach by shifting the focus from data towards decision management. An alternative approach can be found in declarative process modeling which focuses on modeling the minimal business concerns, and leaves as much freedom as permissible to determine a valid and suitable execution plan. Examples of such approach can be found in [9,8]. Declarative process modeling however puts more focus on describing the process constraints.

3 Decision Structures and Modeling

3.1 The Structure of Decision Tables

Decision tables [10,2], enhanced with the semantic richness of SBVR [11] to specify conditions and actions, provide an excellent way to represent decisions, however most real decision cases will not be represented in one single decision table [12]. That is because decisions may involve sub-decisions and lower-level decisions. If we try to fit all this in one table, the table would be too large, too complex, or too difficult to maintain.

Applying structure to a decision table is a way to handle the size and complexity of large decision tables and has always been part of decision table practice [5]. The structure of a decision table is the decomposition of a decision table into a set of interrelated sub-decision tables [4,3] within the same problem scope. Within this structure of interrelated tables, the sub-decision tables derive conclusions for the conditions that are used in their parent table.

Consider e.g. the following (simplified) decision about an individual applying for naturalization (e.g. in Belgium) [13]. Obtaining citizenship is dependent on several requirements and can be obtained in different ways. Some of the considerations are: is the applicant's current residence status appropriate? (multiple possibilities here), can the applicant show that he is socially and culturally integrated?

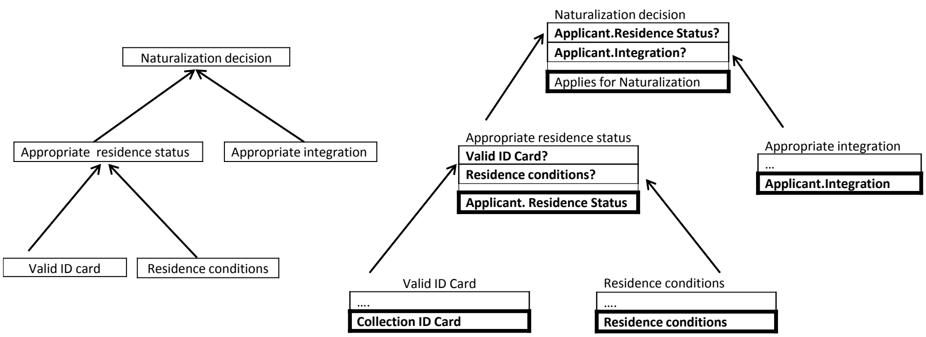


Fig. 1. Structure of a table for a naturalization example

Figure 1 presents two different representations for the structure of a decision table that details when a foreigner can apply for naturalization in Belgium. The left-hand sided structure is a more compact representations while the right-hand sided offers a more detailed representation. The direction of the arrow indicates that the originating table serves as input for the receiving table. The structure of decision tables helps with the modular decomposition of decisions in smaller and lower-level decisions, although they also improve the overview, ease of use, incremental development [14], modularity of rules [15], maintainability [16], etc. The details of each decisions are included in a decision table, e.g. figure 2 details the decision logic of the appropriate residence status table.

Valid ID card	not ok		ok
Residence conditions	not ok	ok	-
Applicant. Residence Status	-	x	x

Fig. 2. Decision table that determines if a residence status is approriate

3.2 The Structure of Decisions

Modeling decision processes without explicitly considering the structure of decisions has some severe drawbacks, because the real decision is then hidden in the process. This can lead to a lack of traceability if the decision structure changes (finding out what the impact of the change is, who is responsible). Additionally, these processes can be hard to maintain, because the decision structure is not available. Especially if the process is (re)designed without explicitly modeling the decision, the process may be implementing an undesirable decision procedure which is the result of historical procedures, responsibilities, roles or structures ('the procedure is like this because the decision has always been taken that way').

Modeling decisions is not (first) about process steps, but about the structure of the decision, independent of the implementation or the execution strategy. Based on existing decision processes and business procedures, we represent decision structures by means of a decision-goal tree or other representations. A decision-goal tree is a partitioning of a decision into subdecisions. It captures the business logic that is needed to make a decision by subdividing the decision. By decoupling the modeling of a decision from the modeling of the process, a decision structure is created that governs the essence of a decision namely the structure and the internal logic of the decision. The concept of a decision-goal tree is similar to a general goal tree [17], but each decision and lower level decision can have a number of extra attributes: data requirements, source of the data, processing time, required lower level decisions in different constellations (all are required, one is required, etc.).

Figure 3 represents a possible decision-goal tree that covers the decision whether an individual applies for naturalization. The goal tree then expresses conjunctions (AND, a single arrow) and disjunctions (OR, multiple incoming arrows) of lower level decisions, although the relations between decisions could be more complex. The internal logic of each (sub)decision is not the immediate concern.

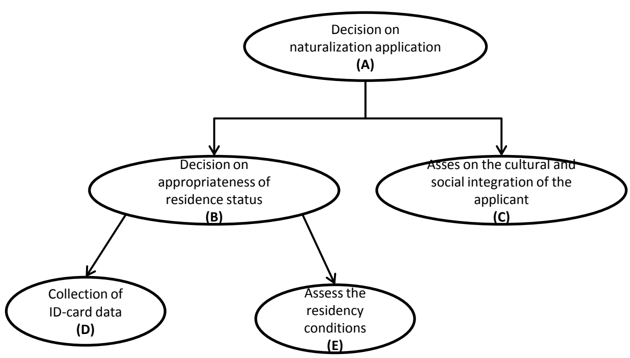


Fig. 3. A decision-goal tree for the naturalization decision

Decision-goal trees are equal to the structure of decision tables if all decisions can be represented by tables. They both give structure to the complicated logic that is inherent to decision, however decision-goal trees don't detail how the decision and sub-decision are implemented while decision tables offer a mechanism that details the implementation of the sub-decisions. The necessity of this distinction arises as not all decisions can be detailed by a decision table and thus a more general concept is needed for the structure of decisions. Although when every sub-decision can be a decision table then the decision-goal tree and the structure of a decision table are equal.

4 From Decision Modeling towards Decision Process Modeling

4.1 Decision Execution Strategies

The decomposition of a decision by a decision-goal tree gives a better understanding of the structure of this decision such that this structure can be used as a guide to develop various execution paths for the decision process. These execution paths could be generated by applying different strategies to the decisions and sub-decisions in the decision-goal tree. Strategies constrain the order in which the sub-decisions are performed, add timing constraints, allocate decision to specific workers, etc. We list the most important strategies:

– The **sequencing strategy** of the subdecisions. The process might attempt to make the decision according to a goal-driven or a data-driven strategy (not unlike reasoning strategies in knowledge-based systems). Data-driven means that once all data for a specific subdecision are available, the subdecision can be executed (even if this later turns out to be an unnecessary path because the goal decision was obtained in other ways). Goal-driven, on the other hand, means that different paths leading to the goal are explored, and subdecisions and data are only examined or requested whenever necessary.
– **Parallelism** of subdecisions. Whenever decisions are executed in parallel, there may be time gains, but if some of the decisions turn out to be redundant, a lot of useless work has been performed.
– Differences in **complexity** of (sub)decisions based on type and outcome. Acceptance decisions (accept/reject) often include large differences in decision complexity or data requirements. Some requirements may be easy, others very hard. If the decision examines acceptance criteria, it might be a good strategy to first evaluate criteria that can be easily rejected because they are not labor-intensive or are easy to answer.

By applying different sets of strategies at a different set of sub-decisions in the decision-goal tree various process models can be generated, offering a wide variety of execution strategies. One example out of the wide variaty of decision process models is presented in figure 4.

4.2 Evaluating Decision Processes

As various process models can result from a decision goal tree, the choice between different process models becomes an important issue. There is a need for criteria to rate the process models such that models can be compared to each other

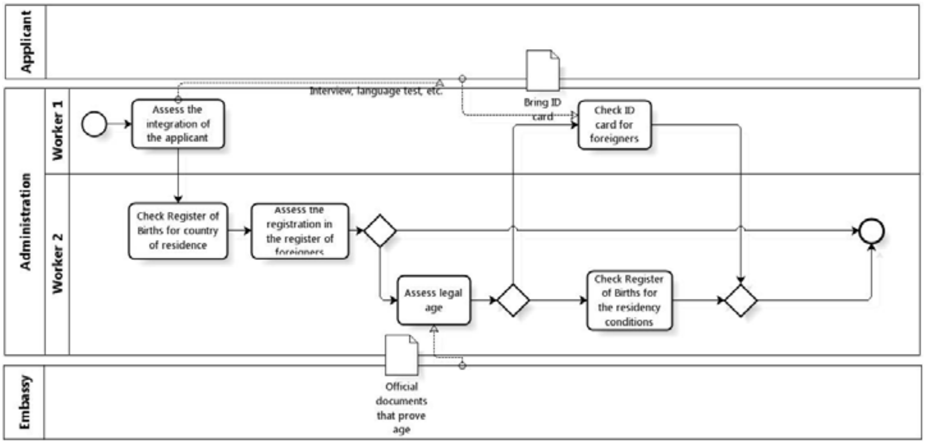


Fig. 4. decision process about naturalization

and the process model that best matches the business strategy can be chosen. Possible process modeling criteria are indicated in [18]. Applying these criteria to the process model in figure 4 already shows some important shortcomings in the process:

- **Customer perspective:** the applicant must come to the administration twice, once for 'assess the integration of the applicant' and once for 'check ID card for foreigners'.
- **Business process behavioral perspective:** starting with a labor-intensive activity such as 'assess the integration of the applicant' is not optimal since the application could be easily rejected when Belgium is not the country of residence (simple check in the Register).
- **Organizational perspective:** the number of handovers can be diminished by letting the 'check ID card for foreigners' succeed the 'assess the integration of the applicant' immediately.
- **Informational perspective:** the current activity sequencing imposes a double consultation of the Register of Births in some cases, while all necessary information could be easily acquired at one point in time.
- **External environmental perspective:** assessing legal age might require intervention from an embassy. The number of contacts with embassies should be limited.

With various decision process models to choose from, the positive and negative aspects of each process model are highlighted by the proposed criteria such that the decision process model that fits best with the business requirements can be chosen.

5 Conclusion

Business decisions are important, but are often not made explicit, hidden in processes or in the head of employees [19]. In fast changing environments, visible decision structures as known from decision table theory and better decision management will allow to create maintainable processes and the adaptation or generation of processes based on strategic criteria.

References

1. Object Management Group: Business Process Modeling Notation (BPMN) – final adopted specification. OMG Document – dtc/06-02-01 (Februari 2006)
2. Moreno, A., Verhelle, M., Vanthienen, J.: Decision tables: An overview of the existing literature. In: Fifth International Conference on Artificial Intelligence and Emerging Technologies in Accounting, Finance and Tax (2000)
3. Vanthienen, J.: Rules as data: Decision tables and relational databases. Business Rules Journal 11(1) (January 2010), http://www.BRCommunity.com/a2010/b516.html
4. Vanthienen, J., Wijsen, J.: On the decomposition of tabular knowledge systems. DTEW Research Report 9604, 1–12 (1996)
5. Codasyl, A.: Modern appraisal of decision tables. Report of the Decision Table Task Group 230–232
6. Browne, E.D., Schrefl, M., Warren, J.R.: A two tier, goal-driven workflow model for the healthcare domain. In: Proceedings of the 5th International Conference on Enterprise Information Systems (ICEIS 2003), pp. 32–39 (2003)
7. Browne, E., Schrefl, M., Warren, J.: Goal-focused self-modifying workflow in the healthcare domain. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences, pp. 1–10. IEEE (2004)
8. Vanderfeesten, I., Reijers, H.A., van der Aalst, W.M.P.: Product-based workflow support. Information Systems 36(2), 517–535 (2010)
9. Pesic, M., van der Aalst, W.M.P.: A Declarative Approach for Flexible Business Processes Management. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 169–180. Springer, Heidelberg (2006)
10. CSA(1970): Z243.1-1970 for decision tables. Canadian Standards Association
11. Object Management Group: Semantics of Business Vocabulary and Business Rules (SBVR), v1.0. OMG Document – dtc/08-01-02 (2008)
12. Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., Baesens, B.: An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. Decision Support Systems 51(1), 141–154 (2011)
13. http://www.kruispuntmi.be/vreemdelingenrecht/wegwijs.aspx?id=84
14. Vanthienen, J., Mues, C., Aerts, A.: An illustration of verification and validation in the modelling phase of kbs development. Data and Knowledge Engineering 27(3), 337–352 (1998)
15. Goedertier, S., Mues, C., Vanthienen, J.: Specifying Process-aware Access Control Rules in SBVR. In: Paschke, A., Biletskiy, Y. (eds.) RuleML 2007. LNCS, vol. 4824, pp. 39–52. Springer, Heidelberg (2007) (Best Paper Award)
16. Vanthienen, J., Wets, G.: Integration of the decision table formalism with a relational database environment. Information Systems 20(7), 595–616 (1998)
17. Letier, E., Van Lamsweerde, A.: Deriving operational software specifications from system goals. In: Proceedings of the 10th ACM SIGSOFT Symposium on Foundations of Software Engineering, pp. 119–128. ACM, New York (2002)

18. Reijers, H., Liman Mansar, S.: Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. Omega 33(4), 283–306 (2005)
19. Taylor, J., Raden, N.: Smart (Enough) Systems: How to Deliver Competitive Advantage by Automating Hidden Decisions. Prentice Hall (2007)

Ontology-Based Approach for Context-Dependent Annotation Templates Recommendations

Cristian Vasquez and Ioana Ciuciu

Semantics Technology and Applications Research Laboratory,
Vrije Universiteit Brussel,
Pleinlaan 2, B-1050 Brussels, Belgium
{cvasquez,Ioana-Georgiana.Ciuciu}@vub.ac.be
http://starlab.vub.ac.be

Abstract. During annotation processes, annotation tools make use of different annotation structures to describe information objects. The chosen structures can depend on the the user needs or the nature of the objects described. In this paper we present the approach that was followed in order to construct an ontology-based matching engine capable of suggesting those structures depending on the context. This paper covers how these annotations were represented in order to follow Linked Data principles, and how Matching Descriptions were used to compute annotation templates suggestions. The design is exemplified in the trust policy specification domain, implemented and used by the annotation tools.

Keywords: Annotation, Template suggestion, RDF, Linked Data.

1 Introduction

Currently, there is no consensus about what an annotation is. One of the most common definitions is that an annotation is an object that describes or says something about other object of information, constituting descriptive information of any type about objects. In this paper we will refer to an annotation as an object that maintains a description about the context, meaning or purpose of an information object, in order to discover, understand and manage that object.

Depending on the domain and purpose of the annotation processes some annotation structures will be more convenient than others, this depends on the specific data artefacts that are going to be annotated and the user contextual information.

Currently, most of the annotator tools available have been designed as domain specific annotator tools, that usually make use of fixed annotation schemes to gather metadata about the annotated resources. In order to enhance the current annotator tools capabilities, we designed an ontology-based engine capable of proposing distinct annotation templates according to the context during the annotation process.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 599–603, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Basically, during the annotation process, the user is aided with suggestions of relevant and valid sets of annotation specifications, which are based on the current interaction with the annotator tool's GUI and NL processing of the text inputs. The suggested specifications are selected according resolved entities from a domain-specific ontology, that specify the constraints and taxonomies used to infer or determine valid annotations according to the domain. Afterwards, these specifications are used to render an input interface to annotate the elements.

This paper focuses on the representation structures and the engine that was used as base for constructing a framework capable of providing this recommendation functionality.

The rest of this paper is organized as follows: Section 1 provides an overview of the approach used to achieve annotation recommendations. In Section 2 a use case is presented. Related research can be found in Section 3. Finally, we conclude the presented work and propose our future work in Section 4.

2 Approach Description

Presenting relevant annotation templates consists in addressing two issues: detecting what is the nature of the information that is going to be annotated, and how this information is going to be annotated.

In order to detect and propose relevant ways to annotate, this approach makes use of two fundamental concepts: "Matching Descriptions" and "Annotation Specs". The former define how to discover entities and properties from an ontology that are good candidates for annotation and the latter determine how these resources are going to be annotated. "Annotation Specs" can also define how the annotation templates are going to be rendered, defining a specific UIs to annotate the resources or alternatively, an annotation syntax to express the annotation.

Also, we need an axiomatized theory (ontology) about the domain, to be able to detect or resolve what ontology entities are going to be annotated using the user partial input, and how this information is going may be validly annotated, using the constraints expressed in the ontology. The ontology was represented using the OWL [3] and RDFS [4] formalisms, published following as Linked Data [5] to promote interoperability.

The followed approach can be divided into two steps: (1) definition of Matching descriptions and (2) Definition of Annotation Specs.

Matching descriptions. A matching description is used to discover which Entities and properties from the ontology are going to be annotated, defining matching algorithms that are going to be applied to such entities and properties. In this approach, this is used to retrieve relevant Annotation Specs according to the user input, making use of a matching engine constructed for this purpose. This engine generates dynamic SPARQL [7] queries used to narrow the results and retrieve RDF representations of Annotation Specs.

These matching descriptions were inspired from a Semantic Matching Framework [8](SemMF) where similarity strategies are used to retrieve information objects expressed in RDF. We used a simpler variant of those matching descriptions, capable of taking into account the user input or context.

Essentially one matching description represents 1) A result, specifying a target resource's class or relations with a target class; 2) Multiple input values, representing a possible user input intended to constrain the search space. 3) variable names constituting handlers to be replaced with values captured from the annotator tools.

The Matching Descriptions can for example state that all the subtypes of a class, e.g. "Person" are allowed as Result, using a defined string matching algorithm over the value of "name", and a range constraint over the value of "age" which is optional, obtaining specific Annotation Specs to annotate persons. Multiple Matching Descriptions can be embedded recursively, for example, to specify how a set of resource's URIs can be retrieved based on their property values, constituting URI resolutors. Values such as optionality or property weights definitions are used to compute ranked result sets via dynamic SPARQL query generation.

Annotation specs. In order to represent one annotation instance, we chose an annotation ontology. We have found that multiple annotation ontologies exist, but only few of them were designed for the annotation of information objects expressed in RDF. Two RDF centric annotation models were selected to perform the annotations, Annotea [1] and the Open Annotation (OAC) Data Model [2]. OAC best fitted our needs since it allows multiple annotation targets, fragments and constrained targets.

OAC basic model allows us to establish links between identified resources that constitutes an annotation. These resources can be for example, resources distributed in the Web. In this model, the annotated elements or resources are "targets" and their descriptions are grouped in a "body", the graph constructed is individualized as an annotation instance and identified with an URI, and constitutes the metadata referred to the "annotated subjects".

Complementary to an annotation instance represented with OAC, an Annotation Spec defines a way to capture the annotation and how the selected resources may be validly annotated. They can be seen as templates used for annotation instantiation following the ontology constraints. These specifications may describe distinct ranges for annotation targets or link to domain-specific body specifications. Also an annotation spec can define a way to capture the annotation describing custom UI, or context-sensitive fragmentations of those UI.

3 TAS^{3}Knowledge Annotator Use Case

The presented approach has been implemented within a tool that we call "Knowledge Annotator" (KA). This tool has been constructed in the context of the EC FP7 Trusted Architecture for Securely Shared Services (TAS^3) project[1], where

[1] http://www.tas3.eu

an ontology based tool is used to aid an user to define valid security annotations at design time. The defined metadata corpus can be considered as a de-coupled layer of security that is used at runtime or compile time to generate code automatically.

The matching engine and API was written in Java following J2EE specifications, constituting a web application deployed into a Tomcat 6 application server. The application uses Jena2 for RDF manipulation and Jena's TDB triplestore for persistence, Where all the elements, like Matching Descriptors, Annotation Specs, annotation instances and the Application configuration are stored to be represented as RDF triples. This provided a straightforward way to share all our data artefacts through the web in a structured way, enhancing re-usability.

This matching engine was integrated using a Web Services API into Intalio BPM[2], an eclipse based modelling tool (IDE) for BPMN models. Here, a set of possible valid annotations are presented to the user at design time, triggered by the selection of elements from the UI or text writing. In this domain-specific tool, the annotations representation was a specific language syntax that was computed directly from the annotation templates expressed in RDF. The syntax is not necessarily known by the user, whereby the tool facilitated the annotation process via dynamic suggestion. Only a valid set of annotations applicable for the selected or written elements are shown, ordered by relevance. Also was up to the user to choose a template to generate annotation instances.

4 Related Work

There are many interpretations of what an annotation is and how an annotation can be constructed or used. Erik Duval et al. [10] provide some guidelines about meta-data principles and practicalities, that are fully compatible with RDF centric annotation schemes such as annotea [1] and OAC [2]. These schemes describes a web-centric light weight structure, that enables annotation sharing across the web infrastructure. Neither annotea or OAC address Frameworks for application-specific annotations.

Regarding RDF matching template techniques, SemMF [8] provides a concise and practical grounding to declare patterns for graph similarity matching, also a formal specification for RDF matching can be seen in Fresnel [9] Selector Languages, used for specifying how to display RDF models. The specific problem of recommendation of templates triggered by the input values is not addressed.

5 Conclusions and Future Work

We have presented an annotator tool designed for suggestion and recommendation of distinct annotation templates, used to annotate domain specific artifact modelled by an ontology.

[2] http://www.intalio.com/bpm

To achieve this, two concepts have been defined, Matching Descriptions and Annotation Specs, used to detect what is the nature of the information to be annotated and how this information is going to be annotated respectively. This convenient separation of concerns allowed us to define simple declarations to constrain and propose valid annotation suggestions taking into account the user context, using a matching engine constructed for this purpose.

This approach have been tested in the "knowledge annotator" framework, constructed as a Annotation recommender for the Security domain, where annotations are suggested following constraints specified in a Security ontology. The implementation is based in RDF representations following linked data principles in order to enhance interoperability.

Future work includes 1) Natural Language bridges for Matching Descriptions generation; 2) A tool and formal model for Annotation Specs definition; 3) Extensions of this framework for recommendation of annotation of complex structures (i.e. multimedia) based on partial selection the content. 4) Testing annotating objects using popular Linked Open Data datasets (or vice-versa).

Acknowledgements. This research is supported by the EC FP7 TAS^3(Trusted Architecture for Securely Shared Services) project. The authors would like to thank all TAS^{3}project partners for their contribution to the research.

References

1. Kahan, Koivunen, M.-R.: Annotea: an open rdf infrastructure for shared web annotations. In: WWW 2001: Proceedings of the 10th International Conference on World Wide Web, pp. 623–632. ACM Press, New York (2001)
2. Haslhofer, B., Simon, R., Sanderson, R., van de Sompel, H.: The Open Annotation Collaboration (OAC) Model (2011)
3. McGuinness, D., van Harmelen, F.: OWL Web Ontology Language - W3C Recommendation (2004/2009), http://www.w3.org/TR/owl-features/
4. RDF Vocabulary Description Language 1.0: RDF Schema - W3C Recommendation (2004), http://www.w3.org/TR/rdf-schema/
5. Tim Berners-Lee, Linked Data (2006), http://www.w3.org/DesignIssues/LinkedData.html
6. RDF Primer - W3C Recommendation (2004) http://www.w3.org/TR/rdf-primer/
7. Harris, S., Seaborne, A.: SPARQL 1.1 query language. W3C working draft (2010), http://www.w3.org/TR/sparql11-query/
8. Oldakowski, R., Bizer, C.: SemMF: A framework for calculating semantic similarity of objects represented as RDF graphs (2005)
9. Bizer, C., Lee, R., Pietriga, E.: Fresnel Display vocabulary for RDF (2005)
10. Duval, E., Hodgins, W., Sutton, S., Weibel, S.L.: Metadata Principles and Practicalities. D-Lib Magazine 8(4) (2002) ISSN 1082-9873, http://www.dlib.org/dlib/april02/weibel/04weibel.html

Maturity Model as Decision Support for Enterprise Interoperability

Wided Guédria[1,2], Yannick Naudet[2], and David Chen[1]

[1] IMS-LAPS/GRAI, University Bordeaux 1. 351, cours de la libération, 33405, Talence cedex, France
[2] SSI, Henri Tudor Public Research Center. 29, Av. John F. Kennedy, 1855 Luxemboug-Kirchberg, Luxembourg
{wided.guedria,david.chen}@laps.ims-bordeaux.fr,
{wided.guedria,yannick.naudet}@tudor.lu

Abstract. To survive in a competitive environment, enterprises have to solve interoperability problems or prevent them to appear, and if possible avoid them by choosing suitable partners having a good interoperability potential. This paper proposes and discusses decision support to enterprise interoperability establishment using MMEI (Maturity Model for Enterprise Interoperability). It discusses how the MMEI model can be used to help enterprises evaluating the suitability of partners in an interoperability context. It also explains how sources of interoperability problems could be identified early during the design phase and so, could be solved before interoperability occurs.

Keywords: Interoperability measure, maturity models, assessment, enterprise interoperability, decision support, best practices.

1 Introduction

Historically, progress occurs when people communicate, share information, and together create something that no one individually could do alone. Moving beyond people to machines and systems, interoperability is becoming a key factor of success in many domains. Interoperability is recognized as the *ability of two or more systems or components to exchange information and to use the information that has been exchanged* (IEEE) [1]. When this ability is not achieved, interoperability becomes a problem that must be solved [2].

In order to survive in a competitive environment, enterprises have to identify interoperability problems that may arise and to deploy solutions to address related challenges. A way to avoid issues is also to choose suitable partners, having a good interoperability potential.

In this paper, we present and discuss the use of the MMEI (Maturity Model for Enterprise Interoperability) [3] as a decision support for enterprise interoperability. After a brief outline of MMEI (section 2), we discuss in particular how interoperability assessment using MMEI provides information that can be exploited both to identify sources of interoperability problems and propose solutions, and to help enterprises evaluating the suitability of partners in an interoperability context (section 3). Conclusion and perspectives are then given (section 4).

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 604–608, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Overview of the Maturity Model for Enterprise Interoperability

Assessing interoperability implies establishing measures of merit to evaluate the degree of interoperability between systems. This is the purpose of so-called maturity models, describing the stages through which systems should evolve to reach higher completeness in the realization of a given objective. Enterprise interoperability maturity can be measured in two ways : *A priori* where the measure relates to the potentiality of a system to be interoperable with a possible future partner whose identity is not known at the moment of evaluation, *A posteriori* where the measure relates to the compatibility between two (or more) known systems willing to interoperate [3].

Based on the Framework of Enterprise Interoperability (FEI) [4] and the main existing interoperability maturity models [5], [6], [7], [8], we have proposed in [3] the Maturity Model for Enterprise Interoperability (MMEI). MMEI deals with the *a priori* measurement of interoperability [3] and addresses the weaknesses of other interoperability maturity models [9]. It considers the three aspects of interoperability [10]: i) *Conceptual* interoperability which is concerned with ensuring that the precise meaning of exchanged information is understandable by any other system that was not initially meant for this purpose, ii) *Technological* Interoperability which is concerned with the technical issues of linking systems and services, iii) *Organizational* interoperability which is mainly concerned with the definition of responsibility and authority so that interoperability can take place under good conditions.

Moreover, MMEI covers the four enterprise interoperability concerns of FEI: i) *Business* level, which deals with how business concerns are understood and shared without ambiguity among interoperating partners; ii) *Process* level, which deals with linking different process descriptions to form collaborative processes and perform verification, simulation and execution; iii) *Service* level, which deals with the capability of exchanging services among partners; iv) *Data* level, which is concerned with the ability to exchange both non-electronic data (documents) and machine transportable data, and use the data/information exchanged.

MMEI defines five maturity levels : 1) *Unprepared*, where there is no capability for interoperation; 2) *Defined*, where there is a capability of properly modeling and describing systems to prepare interoperability; 3) *Aligned*, where there is a capability of making necessary changes to align to common formats or standards; 4) *Organized*, where there is a capability of meta modeling to achieve the mappings needed to interoperate with multiple heterogeneous partners; and 5) *Adaptive*, where there is capability of negotiating and dynamically accommodating with any heterogeneous partner.

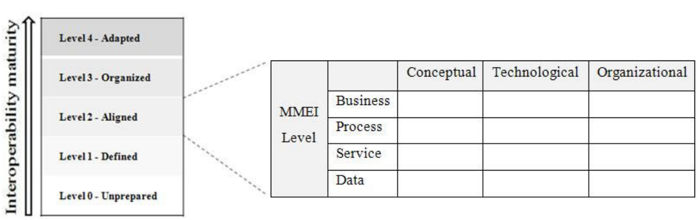


Fig. 1. Zoom on MMEI level

Each maturity level is described by a matrix having two dimensions: “interoperability barriers” and “enterprise concerns” (see figure 1).

A cell in the matrix is called an “area of concern”. Each one contains a description summarizing best practices allowing to remove interoperability barriers.

Best practices are tasks and activities that when put in place, remove interoperability barriers and allow reaching a targeted level of interoperability maturity [3]. They provide principles that stay applicable to a wide variety of enterprises and that remain valid over time. In describing practices, the MMEI focuses on the “what” and not the “how.”

They provide principles that stay applicable to a wide variety of enterprises and that remain valid over time. In describing practices, the MMEI focuses on the “what” and not the “how.” These practices describe the “what’s” in broad terms so that organizations have enough freedom when implementing the “how”. If we take the Business-conceptual area of concern as an example, best practices correspond to the activities and guidelines that should be in place in order to avoid conceptual problems, when business interoperability is concerned. Among these practices, we can cite: the use of standards to define the business model; the availability of possibilities to change the used standard; the knowledge of modifiable elements of the business model.

3 Interoperability Diagnosis and Partner’s Choice Using MMEI

MMEI is intended to be used by assessors; these persons who are in charge of assessing the interoperability potential of an enterprise and to detect areas needing improvements to reach interoperability objectives. Usually, assessors observe the enterprise functioning through series of interviews. The collected data informs on the state of the enterprise and practices in place regarding interoperability. Maturity levels are obtained after synthesis and validation.

Additionally, collected data can be used as a knowledge base to determine interoperability issues during establishment of interoperation with a known partner.

In [2], Interoperability is ontologically defined as a problem to solve, where a problem appears when some existence conditions are met; and is solved by some solution that can induce other problems. Assuming there exist a predefined knowledge base of interoperability problems and solutions; this knowledge base formed during potential interoperability assessment contains knowledge related to existence conditions of some known problems. When the enterprise starts or investigates interoperations with a known partner, some existence conditions might be actually verified depending on the partner structure and functioning. Problems, that will have to be solved for ensuring correct interoperability, are then identified.

For example, at MMEI level 2, the enterprise has to verify if its partner uses compatible standards; whether its models are defined or not; if it has the capacity to change to facilitate business alignment, if it has compatible infrastructure; etc.. Best practices in MMEI are actually used as preconditions to interoperation. Their application allows removing interoperability barriers, and is mandatory at least for one of the enterprise having been assessed with MMEI or its future partner. One of the two at least has to be enough flexible to adapt to the other.

From an enterprise point of view, the knowledge of its MMEI assessment can finally be an important asset in making decisions regarding most suitable partners to work with. This choice is usually not a simple task. It requires investigations about the partner, information about its working procedures, business rules, areas where it is considered as a competitor, etc.

MMEI best practices help simplifying this task. For example, at MMEI level 2, an enterprise has to use standards to facilitate alignment with potential partners. Hence, the enterprise at this level, knows which are its used standards, technologies, and procedures, and is able to know with which of these it can be aligned without problems.

Based on this knowledge, an enterprise is able to design a set of requirements to be verified by a potential partner in order to ensure a correct interoperation. This set can be used to design a stereotype model (e.g. a vector of properties, or an instance of a dedicated ontology) of the suitable partner. Assuming enough information on potential partners is available to build their own models, matchmaking with the stereotype can be achieved using a suitable similarity measure, thus allowing to classify targeted partners according to their suitability.

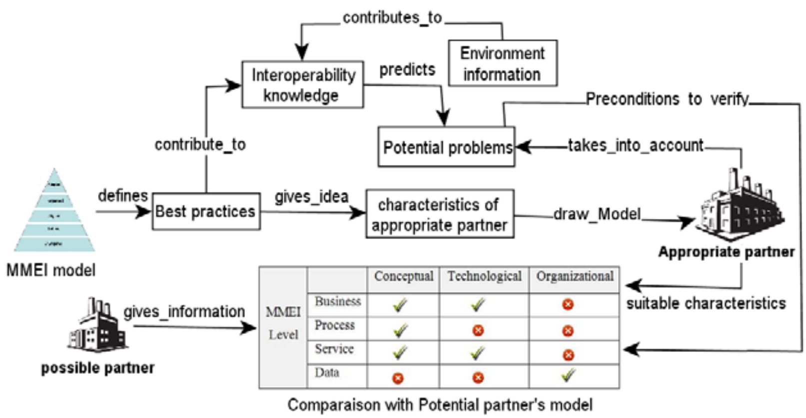


Fig. 2. *A priori* problems diagnosis and choice of interoperability's partners using MMEI

Figure 2 depicts how the MMEI is used to help enterprises in making decisions. First of all, the considered enterprise assesses its potential interoperability using the MMEI model, comparing its state and functioning to best practices specified by MMEI. Adjustments can be made when a targeted level is not fully reached. Next, the list of used MMEI best practices and information that assessors have collected about the enterprise environment (i.e. existing enterprises of the same domain, competitors, ...) give to the enterprise a knowledge about its potential interoperability and an outline about the characteristics of its appropriate partner (i.e. standards that are used and that can be aligned with the enterprise standards, ...). This allows the enterprise to predict potential problems of interoperability and to draw the model of the appropriate partner to interoperate with.

Finally, in the scope of future interoperation, the enterprise collects information about candidate partners and compares their characteristics to the stereotype of the ideal partner. The suitable partner is the nearest one to the stereotype.

4 Conclusion and Perspectives

In this paper, we have presented how the Maturity Model of Enterprise Interoperability (MMEI) can be used as a decision support in interoperation establishment. We have given a brief description of this model and its related best practices. MMEI can be very helpful for identifying a priori interoperability problems and setting up guidelines to choose suitable partners. Such identified future partners are the ones with which the enterprise has less potential problems. It provides a decision support, leaving experts deciding. However, a fully automated system would require a formal modeling of enterprise and the interoperability domain, as well as of problems and solutions related to interoperability. Then, with a formal model of the MMEI and its best practices, practices in place and barriers to interoperability would be identified automatically. The use of systemic theory and formal reasoning with ontologies gives us a possible approach, as shown in [2], [11], which we have started to explore and will carry on by integrating the OoEI [11] with MMEI in a reasoning system.

References

1. IEEE (Institute of Electrical and Electronics Engineers).: Standard computer dictionary, a compilation of IEEE standard computer glossaries (1990)
2. Naudet, Y., Latour, L., Guédria, W., Chen, D.: Towards a systemic formalization of interoperability. Computers in Industry, Special Issue on Integration and Information in Networked Enterprises 61(2), 176–185 (2010)
3. Guédria, W., Naudet, Y., Chen, D.: A Maturity Model Assessing Interoperability Potential. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 276–283. Springer, Heidelberg (2011)
4. CEN/ISO 11354-1, Part 1.: Framework for enterprise interoperability, Draft International Standard ISO/DIS 11354-1 (2009)
5. C4ISR Interoperability Working Group.: Levels of information systems interoperability (LISI), Tech. report, US Department of Defense, Washington (1998)
6. Clark, T., Jones, R.: Organizational interoperability maturity model for c2. In: Proc. of the 1999 Command and Control Research and Technology Symposium, Washington (1999)
7. Tolk, A., Muguira, J.A.: The levels of conceptual interoperability model. In: 2003 Fall Simulation Interoperability Workshop, USA (2003)
8. ATHENA.: Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications, FP6-2002-IST1, Integrated Project Proposal (2003)
9. Guédria, W., Naudet, Y., Chen, D.: Interoperability Maturity Models – Survey and Comparison –. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 273–282. Springer, Heidelberg (2008)
10. ISO 14258, Industrial Automation Systems. Concepts and Rules for Enterprise Models, ISO TC184/SC5/WG1 (1999)
11. Naudet, Y., Guédria, W., Chen, D.: Systems Science for Enterprise Interoperability. In: IESA 2009 Workshops, 5th International Conference Interoperability for Enterprise Software and Applications, Beijing, China (2009)

VADER 2011 PC Co-chairs' Message

Traditionally, handling changing requirements, faults, or upgrades on different kinds of software-based systems have been tasks performed as a maintenance activity, conducted by human operators at design/development time. However, factors such as uncertainty in the operational environment, resource variability, or the critical nature of some systems that cannot be halted in order to be changed have led to the development of systems able to re-plan and reconfigure their structure and behavior at runtime in order to improve their operation without any human intervention.

The goal of our workshop is to explore the features which introduce elements of change in software systems and in service ecosystems. For these purposes, it focuses on adaptive systems, dynamic architectures, and variability management. In general, it intends to determine the synergies among architectural evolution, variation, and adaptation, and the influence of the underlying structure in their self-adaptive properties and attributes. Adaptation is considered as an architecture level feature, and therefore it cannot be dissociated from architectural specification. Variability can be introduced as a natural conceptual evolution of component-based architectures, and presents itself as a significant feature for open, dynamic, and self-adaptive architectures.

Likewise, the intention of this workshop has been to gather software engineering researchers from diverse fields related to the development of variable systems (including systems of systems and families of systems), adaptive architectures, and service ecosystems in order to identify their critical research challenges, as well as to discuss the relevant models, techniques, tools, industrial cases, and methodologies for the development of those complex systems, which share the common feature of being able to dynamically adapt their behavior.

VADER 2011 (the Second International Workshop on Variability, Adaptation and Dynamism in Software Systems and Services) was held in conjunction with On The Move Federated Conferences and Workshops (OTM 2011), during October 17-21 in Hersonissou, Crete, Greece. Though the workshop received a new name and extended its scope significantly, it inherited the tradition and built on the success of a previous edition, AVYTAT 2010 (the First International Workshop on Adaptation in Service Ecosystems and Architectures), continuing its numeration. The most significant addition, with regard to the previous edition, is the inclusion of variability, together with its mechanisms, as an additional element of change.

The workshop had the privilege of counting on a high-quality Program Committee composed of world-wide recognized experts in these distinct but related research areas. Including the Co-chairs, the committee comprised more than 40 researchers from such countries as Denmark, Australia, Spain, Portugal, the UK, France, Finland, the USA, and Belgium. Most of these experts have specialized in one or several of the intertwining research areas of the workshop, namely, the facets of change: variability, evolution, dynamism, architecture or services—and

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 609–611, 2011.
© Springer-Verlag Berlin Heidelberg 2011

ultimately, their relationships and synergies. Their different approaches provided a multidisciplinary set of criteria for the workshop as a whole, defining a unique perspective nonetheless.

The workshop was a success from every point of view and received even more attention than in the previous edition—in fact, all figures increased significantly; in particular, it attracted more than double the submissions of the previous edition. After a thorough reviewing process, in which every paper was revised by at least three (and often even four) members of the above referred Program Committee, it achieved a final balance of a 36% of the submitted articles accepted as full papers, with an additional 18% of articles accepted as short papers.

Accepted technical papers were divided into two groups for their inclusion in these proceedings. The first group deals with variability in software architecture, while the second group deals with dynamic and adaptive architectures, in their different flavors.

Within the variability topic, the paper by J. Diaz et al. describes the process for documenting variability design rationale of flexible and adaptive architectures alongside their architectural description. This process is supported by the meta-models Flexible-PLA and Product-Line Architectural Knowledge, which define the modeling primitives to completely describe the structure of product-line architectures and to document variability design rationales, respectively. Later, the paper by J. M. Moreno-Rivera et al. presents a combination of KobrA and OVM for modeling the variability in the WebGIS Software Product Line. KobrA uses containment trees to specify the relations among components at development time and the composition tree to specify a runtime instance. This approach improves KobrA notation, including OVM, and exploits the global overview of containment trees. These improvements will allow engineers to navigate throughout the system in a clear way, reducing the size of the model and facilitating the verification and testing tasks. Finally, the paper by H. Younessi et al. proposes a framework for building applications that are based on recombining sequences of actions so that the applications' behaviors are entirely specified and tested upfront. With this framework it will be possible to develop trusted behaviors, and trusted adaptations of these behaviors, differently from component-based adaptation strategies.

Within the dynamic architecture group, the paper by T. Giang Le et al. introduces a new language called INI, which can be used to write self-adaptive software through event synchronization and reconfiguration mechanisms. This language therefore combines both rule-based and event-based programming paradigms, by allowing the definitions of rules that can be triggered by events. On the other hand, the work by T. Mikkonen et al. presents a list of architectural issues and derives the initial stages of a reference architecture to serve as a starting point for the design of service mashups. The goal of this work is to facilitate the development and maintenance of mashups, and ease the transition towards Web-based software development. Finally, the paper by S. Prez-Sotelo et al. explores several structural concepts as the basis of an architectural approach to provide self-adaptivity to agent-based systems. The dynamism is supported by an evolving architectural structure, based on combining predefined controls and protocols, in the context of the service-oriented and agent-based architectures. Thus, the agents are coordinated and reorganized by their matching in adaptation patterns, and then evolve into "stable" organizations.

To conclude, we would like to thank the authors for their submissions, the Program Committee members for their excellent reviews, and the OTM organizers for their support and advice in redefining, organizing, and setting up this workshop.

August 2011

Miguel A. Prez-Toledano
Carlos E. Cuesta
Renaud Pawlak
Javier Cámara

A Process for Documenting Variability Design Rationale of Flexible and Adaptive PLAs

Jessica Díaz[1], Jennifer Pérez[1], Juan Garbajosa[1], and Alexander L. Wolf[2]

[1] Technical University of Madrid (UPM) - Universidad Politécnica de Madrid, Systems & Software Technology Group (SYST), E.U. Informática, Madrid, Spain
yesica.diaz@upm.es, {jenifer.perez,jgs}@eui.upm.es
[2] Department of Computing, Imperial College London, London, UK
a.wolf@imperial.ac.uk

Abstract. Variability is a means for evolution of component-based architectures, driving flexible and adaptive architectures. In recent years, researches have emphasized the need for documenting architectural knowledge to maintain and evolve software, i.e. the need for documenting not only the design of the solution, but also the decisions driving the design and their rationale. However, few approaches document the architectural knowledge behind variability, known as variability design rationale. This paper presents the process for documenting variability design rationale of flexible and adaptive architectures alongside their architectural description. This process is supported by the metamodels *Flexible-PLA* and *Product-Line Architectural Knowledge* which define the modeling primitives to completely describe the structure of product-line architectures and to document variability design rationale, respectively. The tool FPLA implements these modeling primitives supporting the specification of architectural models ready to be involved in a model-driven development process. It is illustrated with an example.

Keywords: Variability design rationale, product-line architectures.

1 Introduction

The specification of variability is a backbone in the development but also in the evolution of families of products, widely known as software product lines (SPLs). Architectural descriptions, and especially architectural variability modeling has become a recurring issue on numerous workshops and conferences due to the relevance that architectural variability has in the evolution of component-based architectures, specifically product-line architectures (PLAs) [14,3]. Although more recently, a second recurring issue is the documentation of architectural knowledge [1,4,9,23], i.e. the capturing of the decisions driving the design and their rationale. Both issues provide the basis to address software evolution. The first one drives the description of flexible and adaptive architectures[1], while the

[1] Flexibility is concerned with variability definition at a given point of time, whereas adaptability is concerned with variation over time.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 612–621, 2011.
© Springer-Verlag Berlin Heidelberg 2011

second one provides the insight necessary to guide software evolution. However, both issues present several challenges that still have to be dealt with.

First, most current approaches supporting architectural descriptions [8,6,20] only consider architectural variability specification by modifying the configuration of the PLA in terms of adding or removing components and connections (*external variability*). From our experience in developing SPLs [13], it is also necessary to specify variations inside components (*internal variability*). Several approaches describe this internal variability by using inheritance or aggregation to establish relations between components and their variants [2,10]. These approaches may imply (i) replication of variants if these variants crosscut the architecture, and/or (ii) multiple inheritance and ambiguity problem. To deal with internal variability, this work is based on a previous concept called *Plastic Partial Component (PPC)* [17] that uses the principles of invasive software composition to improve flexibility and reusability of architectures. A PPC is a component with a part of its behavior common for all the products of the SPL, and another one variable for specific products of the SPL. As a result, the variability mechanism underlying PPCs pursues to increase the flexibility and adaptability of architecture design through highly malleable components that can be partially described to implement variability inside components.

Second, few approaches address the documentation of the architectural knowledge behind variability —*variability design rationale*—, i.e. the design decisions driving the variability, their dependencies and rationale. In recent years, researchers have emphasized the need for documenting architectural knowledge to maintain and evolve architectural artifacts [1,4,9,23] and to avoid architectural erosion, drift, or aging [18]. If the explicit capture and documentation of design rationale provide important advantages in software architectures, these advantages are still higher in PLAs because the architecture does not realize the design of a single product, otherwise the design of a family of products. As a result, various stakeholders of various products of a product-line must agree on all design decisions related to the definition of the PLA, for the common and variable parts of the PLA. Therefore, the documentation of variability design rationale has a high strategic value and deep impact on the success of PLAs design, and by extension on the product-line development and evolution [12].

This paper defines the process for documenting the variability design rationale of flexible and adaptive architectures. As variability design rationale should be documented alongside the architectural description, architectural descriptions should support the documentation of knowledge of this sort. Key to this approach is a rich architectural model that combines the architectural concepts that support for internal and external variability with those from the field of architectural knowledge. To support this, two metamodels are required: *Flexible-PLA* and *Product-Line Architectural Knowledge*. The first one defines the modeling primitives to specify external and internal variability of architectures using the concept of PPC. The second one defines the modeling primitives to document architectural knowledge, and specifically variability design rationale. The automation of the process is implemented by the tool Flexible-PLA (FPLA).

FPLA is an graphical tool that implements the modeling primitives that the metamodels before mentioned define, guiding the construction of models and guaranteeing their correctness. FPLA assists software engineers in specifying and documenting architectural models ready to be involved in a model-driven development (MDD) process. The process defined in this paper is illustrated with an example and several snapshots of the tool FPLA.

The structure of this paper is as follows: Section 2 introduces the main concepts of the metamodels Flexible-PLA and Product-Line Architectural Knowledge. Section 3 describes the process for specifying and documenting flexible and adaptive architectures. Section 4 illustrate our proposal using the tool FPLA. Section 5 discusses related work. Finally, conclusions are presented in Section 6.

2 Background

Below, we briefly describe the concepts of the Flexible-PLA metamodel [17] and the PLAK metamodel[2] needed for the purposes of this paper.

2.1 Flexible-PLA Metamodel

The Flexible-PLA metamodel defines the modeling primitives to completely describe the structure of PLAs by specifying external variation, but also internal variations. Internal variations are specified using Plastic Partial Components (PPCs). The variability of a PPC is specified using *variability points*, which hook fragments of code to the PPC known as *variants*. These variants, as well as components and PPCs, realize the requirements at architectural level. We have stated that those variants that realize crosscutting concerns are called *aspects*, and those that realize non-crosscutting concerns are called *features*. Variability points allow us to specify the *weaving* between the PPC and their variants —aspects or features—. The weaving principles of aspect-oriented programming (AOP) [11] provide the needed functionality to specify *where* and *when* extending the PPCs using variants. A formal and more complete definition of the PPC, its metamodel and its graphical metaphor can be found in [17].

2.2 PLAK Metamodel

The PLAK metamodel defines the modeling primitives to capture architectural knowledge including variability design rationale. The modeling primitives are the following: *Closed design decisions* (Closed DDs) support the realization of the common structure of SPLs. *Open design decisions* (Open DDs) support the realization of the variability of SPLs. Open DDs consist of a set of optional design decisions. *Optional design decisions* (Optional DDs) support each of the variants of an Open DD. *Alternative design decisions* (Alternative DDs) support the alternative realization of Closed and Open DDs. *Constraints*, *assumptions*,

[2] The PLAK metamodel is the main contribution of a paper currently under review.

rationale,*design*, and *patterns* are the major elements of design rationale models [22] and so adopted in our metamodel. *Feature concepts* are the concrete representation of requirements in the metamodel. This follows Czarnecki's feature metamodel [5]. *PLA concepts* are the concrete representation of architectural design in the metamodel. This follow the Flexible-PLA metamodel. As DDs act as the traceability links between requirements and PLAs, *Linkage rules* comprise the semantics that establish the bridge between features and PLAs. Linkage rules define the logic to create links between specific concepts of the feature metamodel and specific concepts of the Flexible-PLA metamodel.

3 The Process for Documenting PLAs

This section defines the process for specifying and documenting flexible and adaptive PLAs, taking into account the following issues: (1) Architectural descriptions should support the documentation of variability design rationale. (2) The process should preserve main properties of software architectures. (3) The process should fit into common frameworks for SPL Engineering.

Issue (1): As architectural descriptions should support the documentation of variability design rationale, it is important to select the appropriate architectural description language. It should provide completely support for specify variability of PLAs: external but also internal variability. Key to this approach is a rich architectural model that combines the architectural concepts that support for internal and external variability with those from the field of architectural knowledge. We propose the integration of the metamodels Flexible-PLA and Product-Line Architectural Knowledge (see Section 2). As a result, the approach facilitates an incremental architecting process by using flexible and adaptive components —PPCs— in which all design decisions and rationale driving architectural design are integrally captured and related.

Issue (2): Since the main properties of software architectures —abstraction and encapsulation— must be preserved, the definition of PPCs and the rest of variations must be specified without mixing these variability specifications with the core descriptions of software architectures. The black box view of architectural elements must be also preserved. This is why the process to define PLAs is carried out by several steps according to a model decomposition into several views. These views are: core, service, variability, weaving, derivation, and product [17]. As the goal of this paper is to document architectural knowledge —specifically variability design rationale— alongside PLAs descriptions, we have considered a new view to capture the architectural decisions driving the design of the solution, the dependencies between design decisions, and their rationale.

Issue (3): The process should fit into common frameworks for SPL Engineering such as that defined by Pohl et al. [19]. This framework defines two subprocesses: domain engineering and application engineering. Domain engineering focuses on defining the common platform for all products of the family, while application engineering focuses on developing products through systematic reuse of the common platform. As a result, the process that we propose provides software engineers with different views depending on the subprocess —domain or application— they are addressing.

Therefore, the process to specify and document flexible and adaptive PLAs is defined as follows:

→ *Domain Engineering*

1. In the *Feature View* analysts specify requirements in terms of features: mandatory and optional features, and features groups which consist of a set of alternative grouped features.
2. In the *Core View* architects specify common (mandatory)components, PPCs, ports, connections, and attachments.
3. In the *Rationale View* architects document the decisions driving the design: (i) Closed DDs that trace mandatory features with (mandatory)components or PPCs, (ii) Open DDs that trace features which hook feature groups with PPCs, (iii) the alternatives to Closed and Open DDs, and (iv) the constraints, assumptions, rationale, and patterns for these DDs.
4. In the *External Variability View* architects specify (optional)components, their ports, connections and attachments.
5. In the *Rationale View* architects document the decisions driving the design: Open DDs that trace optional features with (optional)components, their alternatives, constraints, assumptions, rationale, and patterns.
6. In the *Service View* architects specify the interfaces and services that components and PPCs provide.
7. In the *Internal Variability View* architects specify the variability points, variants —features and aspects—, and weavings that constitute PPCs.
8. In the *Service View* architects specify the services that variants provide.
9. In the *Weaving View* architects specify the pointcuts and advices of weavings, i.e where and when extending PPCs using variants.
10. In the *Rationale View* architects document the decisions driving the design: (i) Open DDs that trace feature groups with variability points, and (iii) Optional DDs that trace grouped features with variants —features and aspects—, their alternatives, constraints, assumptions, rationale, etc.

    Therefore, for the purpose of the domain engineering process, Closed DDs are completely closed (or bound) and Open DDs are intentionally left open (or delayed) to support the customization of the products of SPLs.

→ *Application Engineering*

11. In the *Derivation View* architects specify the configuration of PPCs for a given product of the family, i.e. variability points are resolved for a specific product of the family, and also indicates those optional components, connectors and attachments which are mandatory for the product.
12. In the *Rationale View* architects bound Open DDs to the appropriate Optional DD.

13. In the *Product View* architects can verify the result of applying the selections of the derivation view. It extends the core architecture by providing the final architecture description of a given product of the family.

    Therefore, the derivation of products implies the binding of all Open DDs to the appropriate options, i.e. their transformation into Closed DDs.

Summing up, flexible and adaptive PLAs are defined by specifying the components —mandatory and/or optional—, PPCs, ports, connectors, attachments, interfaces, services, variability points, features, variants, and weavings. Then, PLAs are documented by specifying the {Closed, Open,Optional,Alternative} DDs, as well as their constraints, assumptions, rationale, and patterns.

4 Illustrating the Process

This section describes a scenario to illustrate the use of our approach using the tool FPLA[3]. The scenario exemplifies a SPL for banking systems. Banking systems typically consist of a set of core features that offer their functionality to ATM machines. From all the functionalities banking systems provide, this example focuses on that of maintaining *accounts balance*. This functionality must fulfill the non-functional requirement of *availability*. Some products of this SPL require *strict* 24/7 availability, while others allow a more relaxed, *non-strict* availability. Following the process described in Section 3, analysts specify the requirements in terms of features. Fig. 1 shows the feature model for this example in which we have considered the following features: *ATMFeature*, *AccountBalanceFeature* and *Availability24/7Feature*. All of them are mandatory as it is represented by fulled circles. The feature *Availability24/7Feature* has a XOR feature group in which only one of the grouped features can be selected to derive a specific product: strict or non-strict availability.

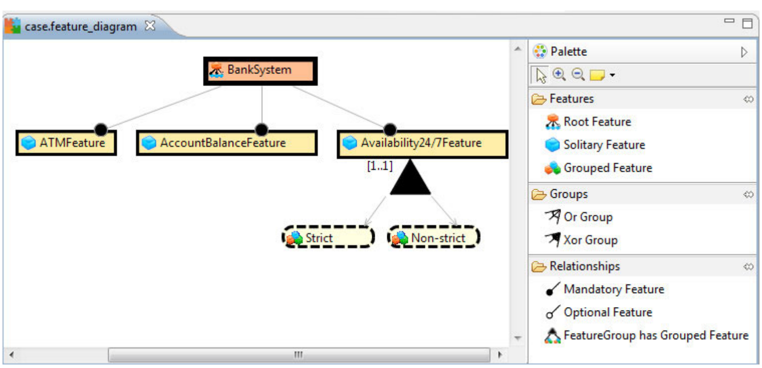


Fig. 1. Banking systems SPL - Step 1: feature model

[3] https://syst.eui.upm.es/FPLA/home/

Following the process described in Section 3, architects should specify the core view of the PLA, i.e. the realization of the feature model at architectural level. Fig. 2 shows the core view in which the component *ATM* and the PPC *Balance* are connected by the connector *c*. At that moment, the architects made several design decisions such as follows. As the architectural element implementing the feature AccountBalance should implement availability in its two variants, this feature is realized by a PPC called Balance which support this variability (see ClosedDD_001 in Fig. 4). Availability is a concern that could crosscut the architecture, i.e. it is a non-functional requirement that could be reused by more than one components in a complete scenario of a SPL for banking systems. As aspects facilitate reusability, the architects decided to implement availability using aspects (see OpenDD_002 in Fig. 4).

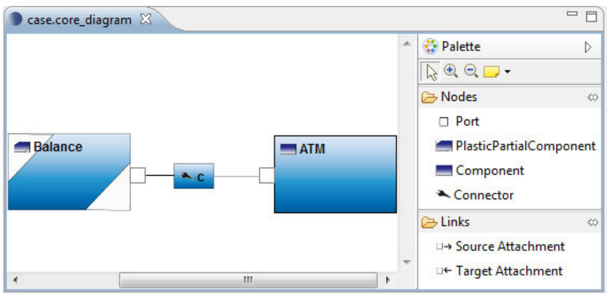


Fig. 2. Banking systems SPL - Step 2: core view

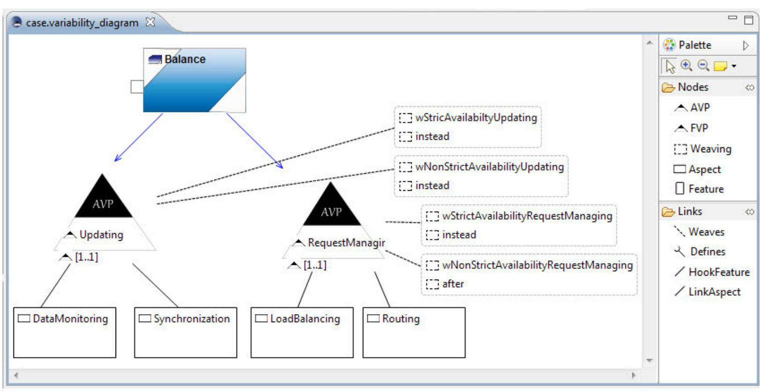


Fig. 3. Banking systems SPL - Step 7: internal variability view

Various architectural tactics to realize availability have been proposed [21]. The architects selected *active redundancy* and *passive redundancy* tactics to implement strict and non-strict availability, respectively, as follows: Fig. 3 shows the internal variability view in which the PPC Balance defines two variability points, *Updating* and *RequestManaging*, and the aspects that implement active and passive redundancy. Then, the weavings that specify where and when extending the PPC Balance using the aspects, should be defined. Fig. 4 shows the design decisions driving these solutions, including variability design rationale (see OptionalDDs ActiveRedundancy and PassiveRedudancy in Fig. 4). Finally, in the Derivation View, the product architect resolve the variability points to derive a specific product. In this scenario, the architect could select the weavings to extend the PPC Balance to implement strict or non-strict availability[4].

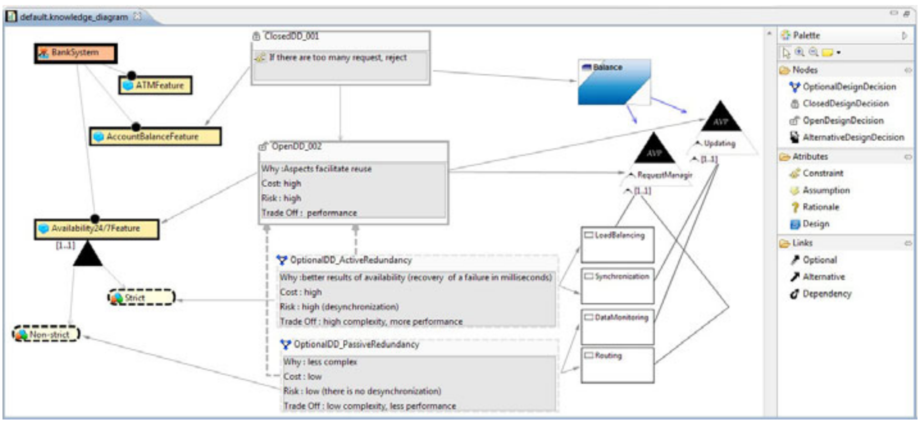


Fig. 4. Banking systems SPL - Step 3, 5 and 10: architectural knowledge view

In this scenario a PLA has been documented applying the process and using FPLA. Once the model has been completed, FPLA supports the automatic generation of a domain-specific language associated to the architectural model built. It is contained in an XML file and can work as input for many different tools with different purposes such as code generation or change impact analysis.

5 Discussion

Moahn and Ramesch [15,16] assert that the documentation of design decisions associated with variations and the capability to trace the life of these variations are key to effectively support evolution of SPLs. In [7], the authors propose a grammar to specify design rationale, including variability design rationale. However, the concept of variability design rationale is very shallow in the sense that it does not include the knowledge about how and why this variability is implemented in the architecture. These works are very close to ours, but they admit that their approaches should be complemented by specialized design-modeling representations to model variation points. In our proposal, aspects

[4] Space restrictions does not allow to show the snapshots for all views.

and features only specify the behavior to be inserted into the PPC, i.e. they do not specify the pointcuts as AOP does. The pointcut is specified outside the aspects and features. As a result, aspects and features are unaware of the linking context, and they are completely reusable. This makes architectures more flexible and adaptable. This along with the documentation of variability design rationale facilitates the evolution and maintaining tasks.

6 Conclusions and Further Work

This paper defines a process that integrates the documentation of variability design rationale as a view into a view model for describing PLAs. As a result, this paper contributes a novel architecture evolution approach, with a tool support called FPLA, which brings together architectural specification and documentation with the aim of capturing, managing and guiding architectural evolution. The tool FPLA supports the definition of both external and internal variations of the PLA configuration, and the documentation of variability design rationale. It provides architects a model decomposed in several views that makes the PLA description more intuitive and friendly. Since FPLA is based on two metamodels and their PLA descriptions are models, FPLA is ready to be involved in a MDD process to transform its outputs into platform dependent models and to automatically generate code. As future works we plan to generate automatically C# and Java code from FPLA models. We also plan to integrate FPLA with a middleware support in order to execute the generated code from models.

Acknowledgment. This work has been partially sponsored by TIN2008-00889-E and TIN2009-13849 projects, and by UPM (Researcher Training program). Authors are indebted to David Musat for participating in the development of FPLA.

References

1. Ali Babar, M., Dingsyr, T., Lago, P., Vliet, H.v.: Software Architecture Knowledge Management. Springer, Heidelberg (2009)
2. Bachmann, F., Bass, L.: Managing variability in software architectures. In: SSR 2001: Proceedings of the 2001 Symposium on Software Reusability, pp. 126–132. ACM, New York (2001)
3. Chen, L., Ali Babar, M., Ali, N.: Variability management in software product lines: a systematic review. In: SPLC 2009: Proceedings of the 13th International Software Product Line Conference, pp. 81–90. Carnegie Mellon University (2009)
4. Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., Stafford, J.A.: Documenting Software Architectures: Views and Beyond, 2nd edn. Addison-Wesley Professional (2010)
5. Czarnecki, K., Helsen, S., Eisenecker, U.W.: Staged configuration through specialization and multilevel configuration of feature models. Software Process: Improvement and Practice 10(2), 143–169 (2005)

6. Dashofy, E.M., Hoek, A.v.d., Taylor, R.N.: A comprehensive approach for the development of modular software architecture description languages. ACM Trans. Softw. Eng. Methodol. 14(2), 199–245 (2005)
7. Galvão, I., van den Broek, P., Akşit, M.: A model for variability design rationale in spl. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ECSA 2010, pp. 332–335. ACM, New York (2010)
8. van der Hoek, A., Mikic-Rakic, M., Roshandel, R., Medvidovic, N.: Taming architectural evolution. In: Proceedings of ESEC/FSE-9, pp. 1–10. ACM, New York (2001)
9. Jansen, A., Bosch, J.: Software architecture as a set of architectural design decisions. In: WICSA 2005: Proceedings of 5th Working IEEE/IFIP Conference on Software Architecture, pp. 109–120 (2005)
10. Kakarontzas, G., Stamelos, I., Katsaros, P.: Product line variability with elastic components and test-driven development. In: Proceedings of CIMCA 2008, pp. 146–151. IEEE Computer Society, Washington, DC (2008)
11. Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, W.G.: An Overview of AspectJ. In: Lee, S.H. (ed.) ECOOP 2001. LNCS, vol. 2072, pp. 327–353. Springer, Heidelberg (2001)
12. Knodel, J., Muthing, D.: The Role of Rationale in the Design of Product Line Architectures - a Case Study from Industry. In: Rationale Management in Software Engineering, pp. 237–254. Springer, Heidelberg (2006)
13. Magro, B., Garbajosa, J., Pérez, J.: Applied Software Product Line Engineering. In: Applied Software Product Line Engineering, pp. 173–199. Taylor & Francis (2009)
14. Matinlassi, M.: Comparison of software product line architecture design methods: Copa, fast, form, kobra and qada. In: Proceedings of ICSE 2004, pp. 127–136. IEEE Computer Society, Washington, DC (2004)
15. Mohan, K., Ramesh, B.: Managing variability with traceability in product and service families. In: Proceedings of HICSS, pp. 1309–1317 (2002)
16. Mohan, K., Ramesh, B.: Tracing variations in software product families. Commun. ACM 50, 68–73 (2007)
17. Pérez, J., Díaz, J., Soria, C.C., Garbajosa, J.: Plastic partial components: A solution to support variability in architectural components. In: Proceedings of WICSA/ECSA 2009, pp. 221–230. IEEE (2009)
18. Perry, D.E., Wolf, A.L.: Foundations for the study of software architecture. SIGSOFT Softw. Eng. Notes 17(4), 40–52 (1992)
19. Pohl, K., Böckle, G., Linden, F.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer, Germany (2005)
20. Razavian, M., Khosravi, R.: Modeling variability in the component and connector view of architecture using uml. In: Proceedings of AICCSA 2008, pp. 801–809. IEEE Computer Society, Washington, DC (2008)
21. Scott, J., Kazman, R.: Realizing and refining architectural tactics: Availability. Tech. rep., Technical report CMU/SEI-2009-TR-006 ESC-TR-2009-006
22. Shahin, M., Liang, P., Khayyambashi, M.: Architectural design decision: Existing models and tools. In: Proceedings of WICSA/ECSA 2009, pp. 293–296 (2009)
23. Tang, A., Avgeriou, P., Jansen, A., Capilla, R., Ali Babar, M.: A comparative study of architecture knowledge management tools. Systems & Software J. (2009)

Evolving KobrA to Support SPL for WebGIS Development

Juan Manuel Moreno-Rivera[1,2], Elena Navarro[3], and Carlos E. Cuesta[4]

[1] Earth Observation and GIS Department, University of Castilla-La Mancha, Albacete, Spain
[2] SIGTel Geomática, Centro I+D de Empresas – PCYTA, Albacete, Spain
JuanManuel.Moreno@sigtel.es
[3] Computing Systems Department, University of Castilla-La Mancha, Albacete, Spain
Elena.Navarro@uclm.es
[4] DLSI II, Rey Juan Carlos University, Spain
carlos.cuesta@urjc.es

Abstract. SIGTel is a SME aimed at developing WebGIS application. It has a portfolio of products which are a combination of Open GIS components and its own developed components. Although most of these products share a common architecture and common features, they have to be customized according to the user requirements. Now, this SME is working at the improvement of the software development process by introducing a Software Product Line (SPL) approach to automate the process of development and deployment based on that common architecture, as well as reduce time to market and improve quality. The first step has been taken, and KobrA approach has been chosen as the basic methodology to start working. Now, the on-going work is to perform the Domain Engineering, being the management of variability one of the first issues to be resolved. Here, we present a combination of KobrA containment tree and Orthogonal Variation Model (OVM), as the selected alternative to model the variability by using the architecture of our WebGIS SPL.

Keywords: KobrA, SPL, WebGIS, component-based development, open GIS.

1 Introduction

The complexity of software development is increasing nowadays, and the competition between enterprises is harder than ever. This situation becomes worst when we get into the context of Small and Medium Enterprises (SME). In this context, SIGTel Geomática [18] portfolio is based on the production chain provided by the Open GIS community, where open source components are combined with their own ones. It has been detected that a high percentage of the development of new products is repeated along the whole production chain and that the 90% of software development projects share a common architecture. This philosophy fits perfectly with the approach of Software Product Lines (SPL). Considering that situation, an SPL-based approach has been chosen, intending to help us to automate the software development process of WebGIS applications [11] as far as possible, as well as managing variability and traceability.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 622–631, 2011.
© Springer-Verlag Berlin Heidelberg 2011

After this first step, an evaluation of existing approaches and tools has been done, deciding on the KobrA approach [12] as the final choice. KobrA was selected to guide the development of the SIGTeL SPL due to several reasons. First, the KobrA philosophy suits almost perfectly to the WebGIS development process because it is a component-based approach, and WebGIS development focuses on components as their basic unit of development. Thus, a component-based approach, as KobrA [3], makes the models easier to understand to WebGIS developers. This approach also takes into account third party components, in the form of plug-ins within the model.

An improvement to the KobrA containment tree is presented here, analyzing which are the aspects of KobrA that needs to be customized in order to be applied in the WebGIS domain. In addition, we present a new approach using KobrA tools combined with Orthogonal Variability Model (OVM) [13] notation. Our proposal reduces the amount of work to be done when solving variability and makes more easy application engineer work. We also introduce a prototype tool developed to support the work presented here.

In this work, Section 2 presents the domain and the basic concepts. Section 3 describes the related works. Then, in Section 4, the application of KobrA methodology within our domain is analyzed by applying it to a case study, the core components of the SPIDER system [10], in order to establish the advantages and disadvantages of this approach within a real world project. Next, after identifying which are the improvements and/or adaptations that this approach needs in order to be applied in the context of WebGIS applications, our proposals is described in Section 5: the adaptations of the KobrA containment tree to improve the variability management and Section 6 presents the case study. Finally, in Section 7, the next steps of our Software Process Improvement (SPI) roadmap, based on the results here presented and the expectations of SIGTel, are described.

2 WebGIS Domain

WebGIS is the acronym for Web-based Geographical Information System. WebGIS's central feature is a map along with a system to analyze, edit and produce spatial information. These systems are becoming more popular every day. Nowadays, just about everybody uses WebGIS, Google Maps© [5] being the most popular example.

Market changes have led to an explosion of the demand with a lot of large and small companies fighting for its place in this emerging market. New users are becoming more familiar with WebGIS and are requesting new features. Complex desktop processes are being adapted to a web-based system as new technologies become available. Besides the increasing complexity of WebGIS, the Open GIS community is growing and offering more possibilities for developing new products. Software development companies are now expected to develop new WebGIS systems in a shorter time with better features and improved quality.

This context makes *reuse* into an even more difficult task because each component needs to be adapted and be more flexible for demands of new features. These difficulties are more critical for small companies, such as SIGTel, that not only have to maintain the production levels but also keep up to date with new market trends. However, the development time of final products is too long and the quality is often not what users expect because of the inherent complexity for testing WebGIS applications. This complexity remains in spite of the reuse of existing components, the development based on the adaptation of existing systems, and the use of widely used communications standards defined by the Open Geospatial Consortium (OGC) [15]. Another problem of WebGIS is that as systems increase their complexity, it becomes a hard task to manage the whole architecture in an efficient and productive way. This problem becomes worst when the phenomenon of *cloning* [4] emerges making more difficult the maintenance of deployed systems and the reuse of new components for specific customers. Finally, this complexity is also related to the difficulties for implementing test benchmarks for systems with so many interconnected components from different providers. Therefore, it is necessary to include some methodological and systematic approach for producing, deploying and maintaining this kind of systems that enable to share a common architecture and be able to manage the variability in an efficient and a more productive way.

3 Related Works

In WebGIS development there is a concept called *GeoStack* that also uses, similarly to SPL, a common architecture along with a family of products. Many enterprises and research groups work around a framework or combination of components that facilitates the adaptation based on customers' requirements of a limited set of optional features for their final products. The problems of this approach are three: (i) it is strongly linked to the technology these components have been developed for; (ii) the variability is constrained to a reduced group of final features included in the systems; (iii) the cloning phenomenon arises and due to the lack of traceability, each component needs independent maintenance efforts. Nowadays SIGTel's GeoStack works like that. The ideal solution would rely on managing the variability in a way that facilitates the assembly of different combinations of components according to specific customers' requirements. For example, if our family of products is based on Flash technologies and a new customer asks for a product without Flash, due to any kind of security or corporate decision, could we provide a solution based on our common architecture that supplies the same features but using non-Flash components? Although we could answer positively this question, the time needed to adapt this architecture to the customer's needs would too long and would increase the final cost of the product, as well as the time to market. Therefore, our company would not be competitive for managing this kind of situations. This situation led us to use SPL in SIGTEL.

SPL approaches can be classified into two different categories: *feature-based* and *component-based* approaches. Feature-based approaches use features as the basic unit and component-based ones focus on the component level. The first ones are more mature and have been used for the last twenty years. FODA [8] settled down the bases for this family of approach in the 90's. However, in WebGIS domain component is the basic unit of development, so component-based approaches, like KobrA or CoPaM [1], fit better to the domain of WebGIS.

No matter which approach is followed, all of them aim at providing suitable techniques to *manage variability*. KobrA, for example, uses UML stereotypes to manage variability. It works but it still needs improvements as explained in Section 5. There are notations, like Common Variability Language (CVL) [16] or OVM [13] that have been considered to improve variability management in KobrA. Both of them have an Eclipse plug-in available, Eclipse CVL plug-in [17] for CVL and VarMod-Editor [20] for OVM, which makes them more suitable to our needs, as this is the IDE used in SIGTel. CVL is mainly used to define feature diagrams. OVM is a standard notation focused on supporting variability in SPL for improving the testing and formal validation of models. This is the reason why the latter has been finally selected, as the proper definition of the testing process is one of the main problems to be resolved in the near future of SIGTEL.

4 KobrA in a Nutshell

KobrA [3] is a model-driven approach for component-based development that can be used as well for deploying SPLs. It is based on the concept that a system, and any element that makes it up, can be modelled as a component. This approach establishes a model that represents the interaction with the external environment and details the internal design of each of the components in a tree hierarchy. The KobrA modelling process of a component entails two different tasks: the *specification* of how a component interacts with the rest of the system and what does; and the *realization* of how the component performs this specification by means of its design. Therefore, KobrA models components in a similar way to WebGIS development approaches because these two different views resembles the way components are assembled in a WebGIS application. In WebGIS, components are connected through their APIs or by standard communication channels with other components or external elements and then are customized according to customer needs.

KobrA uses two tree structures: the *containment tree,* to specify the relationships between components at development time and the *composition tree,* to specify a run-time instance of an existing system. The main idea is that both of them are aligned so that it is easy to define a relationship between the design and the runtime execution of new systems. Considering these two notions of composition, KobrA allows us to model both the design and the execution of our systems, which can be very useful in order to define a testing approach for this kind of systems. The containment tree is used during the *domain engineering phase* to represent the structure of the system, that is, its design and the components distribution. This tree is instantiated during the *application engineering phase* to deploy different products by resolving the existing variation points. Afterwards, each instantiated tree is refined and transformed into another tree structure to represent the logical and physical distribution of the specific product following the *KobrA flattening process*, and its modelled components are replaced by code components applying the *KobrA translation process*.

During the application engineering phase, the engineer goes throughout the containment tree solving the variability starting from the top node, which represents the whole system, to the leaf component nodes, till all the variability is solved for each specific product. The first consequence of this approach is that there is more redundancy and verbosity than necessary. due to the *localized view* of each component. The reason is that KobrA establishes a set of models for each component, to ensure its independent and homogeneous treatment. However, the number of models required for modelling components and the verbosity along the tree, turns both variability resolution and models maintenance into two error-prone and tedious tasks. In addition, the decision models used to resolve variability are text-based, and this makes the process too complicated, even with tool support. For this reason, it would be worthwhile to provide engineers with a means to deal with the variability in a more agile way during the application engineering phase. In addition, reducing the number of models and diagrams will also reduce the efforts needed to maintain and update the framework of our WebGIS SPL.

In addition, there is a lack of automation in KobrA, especially traumatic at the implementation phase as there are no defined methods or processes to transform design models to implementation models. Finally, there is no tools support for KobrA approach. By the time this work has been written, the KobrA developers are working at the development of a supporting tool for KobrA approach. However, this tool is not available so that it was not possible to test and validate it. For this reason, we have had to develop our own prototype, based on GMF [6], to support the modelling of KobrA compliant components.

5 Evolved KobrA Containment Tree for Supporting WebGIS Domain Engineering

KobrA approach fits perfectly with the way WebGIS systems are designed. But it is a general purpose approach and some customization needs to be applied to improve the performance of this approach. For example, as pointed out in [3], KobrA context realization, a.k.a. domain engineering, is focused on Enterprise Resource Planning (ERP) domains. KobrA uses package containment of UML to model the architecture and the hierarchy of components of a system. It also models the relation, run-time and design dependencies, between components from different branches of the tree but in this components tree no variability is modelled. As mentioned before, this means that application engineers need to go along the containment tree, node by node, resolving all the decision modelled for each node. By locally managing variability, as KobrA does, engineer loses the global view of the system offered by the containment tree. Apart from that, variability dependencies are only managed through the tree hierarchy from the components tree which is not enough to manage horizontal variability dependencies. More than that, decision models are text-base, which is not very helpful during application engineering process. Finally, variability is only used to manage the inclusion of features, but there is no variability management during implementation phases which offer an opportunity of choosing between different implementations of components, to provide a more adaptable architecture for SIGTel's SPL.

In the WebGIS domain, *variation points* emerge at the components level, while designing new instances of our system, because it is at this moment when engineers decide which of the existing physical components are chosen to implement selected features. For third party components, getting inside each component node is not necessary because most of them are integrated in members of our family of products as they are provided. In fact, in the WebGIS domain there is an initial variability decision process that focuses on deciding which components are going to be included to implement features of new systems taking into account the user requirements. Therefore, the enhanced containment tree presented here should allow engineers to navigate throughout the system in a simplified and clear way, identifying in every single moment how components and variation points are connected and their dependencies during the application engineering phase. It would also facilitate reducing the number of models necessary for each component, so that the interaction with external components would be modelled in this enhanced containment tree, and each node would focus on the realization of its elements. Also, third party components would not need, in most cases, an internal definition because they would be used as they are provided by just considering their APIs.

The KobrA notation for variability uses the tag <<variant>> together with components and attributes to model each node of the containment tree by means of class diagrams. We propose an enhancement to improve the KobrA notation and exploit the benefits of the global overview of the system provided by the containment tree. This enhancement will allow engineers to interact with it while maintaining the SPL reference architecture and while deriving new products during the application engineering. The chosen notation to develop this enhancement is based on OVM.

In the context of WebGIS, a containment tree can become very huge so that if an application engineer wants to instantiate this model following KobrA guidelines, he/she has to go throughout the components hierarchy. This can be a very hard and tedious task even by using a modelling tool or a model-driven process. Here one of the problems of the component-based approaches arises, as described in [12]. By improving the variability notation of the containment tree, the fast prototyping skills of the SPL approaches based on KobrA are promoted. In [3], it is established that "there is no reason to support variability anywhere in a composition tree unless they have an effect on the properties of the overall system represented by the root". The idea is that KobrA manages the variability of the whole system at top of the containment tree. However, with the enhancement proposed, the work to be done

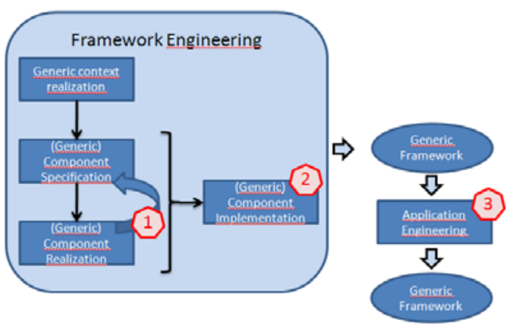


Fig. 1. Product Line life-cycle according to the KobrA Approach [3]

during the creation, maintenance and update of the SPL framework can be reduced (see point 1 at Figure 1). In addition, this enhancement facilitates the application engineering tasks by making more agile the flatting and translation process of the KobrA approach by using OVM notation (see point 3 at Figure 1). By considering the variability in the containment tree, the adaptability of our SPL can be improved. As our WebGIS SPL is based on Open GIS components, it is necessary to manage not only which features should be included or not, but also which component should be used for implementation (see point 2 at Figure 1). These decisions can be based on user requirements, expertise of the engineer or even changes in the community of any of the third party components. We need to make our SPL more adaptable so that it does not depend on a concrete component and its implementation, because the open software community is one of our providers, and therefore our SPL should not depend on any specific component.

6 Case Study

As mentioned before, our case study is SPIDER [10]. Here, a reduced version of this system is presented focusing on its core elements. It is a good example of the architecture shared by many WebGIS projects developed in SIGTel, but at the same time it represents one of the more complex systems that can be developed. This can be considered a good example because even only with the core components, it is complex enough to be considered a real example.

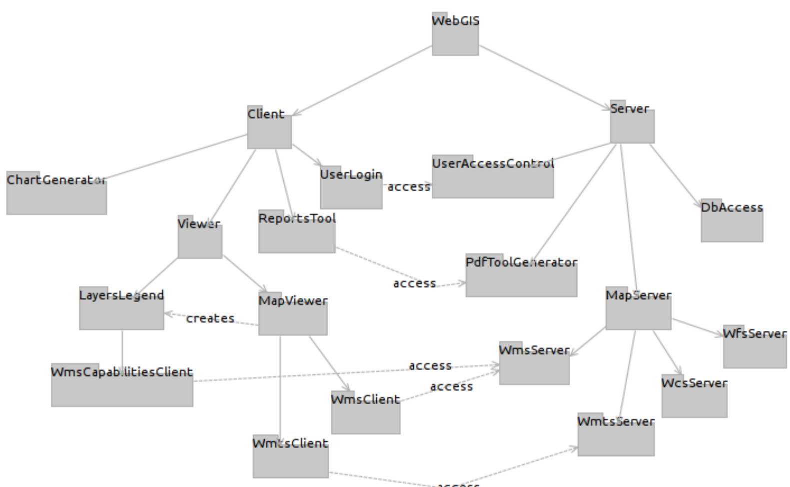


Fig. 2. SPIDER core components architecture with KobrA

Figure 2 illustrates a containment tree of the core components of SPIDER architecture modelled with the previous version of KobrA components tree. It shows in a glance the whole system architecture. As can be observed, it is not possible to identify the relationships of variability (following the KobrA approach) without getting into each component, where UML diagrams model the *localized* variability for each node of the tree. Furthermore, KobrA relationships between components do not consider the <<excludes>> relationship, which is critical for managing variability, as far as this relationship is basic for variability management. One component removal from the system can make other components incompatible with the instantiated version of the architecture. The only way it can be specified is by including in the decision model of each node which components or features have also to be removed. However, this is a difficult task as these decision models are text-based, making a difficult task its inclusion. This is another reason to include OVM because this notation considers this relationship. As can be observed in Figure 3, it facilitates to easily identify the Variation Points existing in your containment tree, what is not the case for the previous KobrA version. Finally, one important contribution of using OVM is that it does not only also consider relationships between components, but also between Variation Points. There is no notation for variability in such an advanced way in the KobrA basic approach. Thus, an improved version of KobrA offers a most suitable philosophy SPL approach for WebGIS, but improving its usability for engineers in a real context as in SIGTel.

Figure 3 shows how now the Variation points “ChartGenerator” and “ReportTool” are linked, so that by solving the variability of one of them implies to solve the variability of the other, depending on the option chosen. At the same time, we not only specify the variability of the components and features to be included, but also of implementation alternatives using different components. In addition, the application engineer can solve the variability of logical functionality and part of the flatting process at the same time by means of this model. For example, the MapViewer now is a variation point and we can start deciding on features and code component in the same process, reducing part of the required process in KobrA. Also, using this approach, it is not necessary to describe detailed models of the third party components, as the engineer is only interested in the dependencies they have and the basic configuration for their deployment.

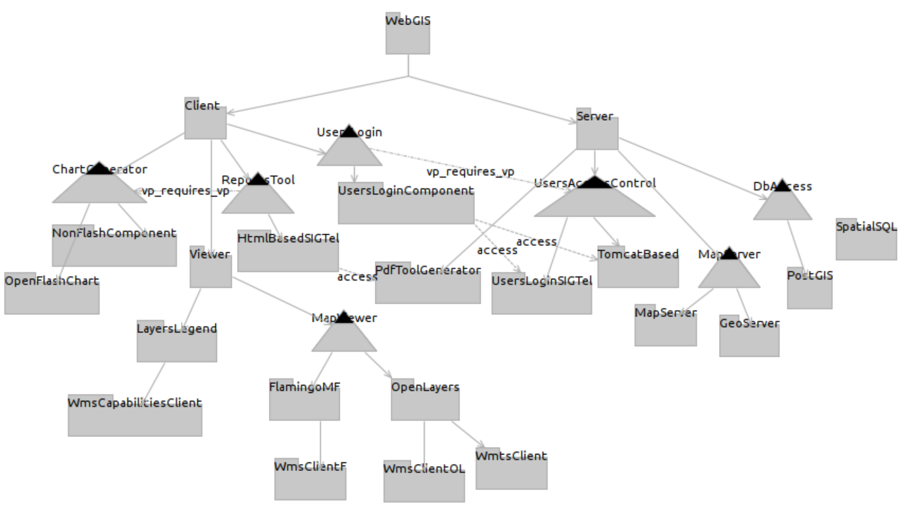


Fig. 3. SPIDER core components architecture with KobrA + OVM

6.1 Tool Support

As pointed out in [2], there is no tool support to ensure that the models are compliant with KobrA. The existing UML modelling tools cannot be used to model any system based on the KobrA approach. The group that created KobrA is working on the development of a tool to support this approach, however it is not available and just some examples of models done with this tool can be found in [2]. Therefore, SIGTel had to start working on the development of a plug-in to support the modelling tasks that were performed during this project, in order to avoid doing all the models manually, which is very tedious and time consuming.

The tools that have been considered to support the development of this work are Eclipse-based tools / frameworks, like MOSKitt [14] or VarMod-Editor. TOPCASED [19] was also considered as it offers an interesting framework for critical and embedded systems, but for this work GMF [6] was used to develop the prototype tool for this work, as it is the main tool to develop graphical modelling tools in Eclipse.

7 Conclusions and Future Work

After evaluating KobrA by means of our case study, we decided to improve its variability management in order to improve its potential benefit for WebGIS SPL development, as well as to offer support for a more adaptable SPL. Combining KobrA containment tree with OVM makes the process clearer and the models more maintainable and reusable, giving an overview of the whole architecture variability in a glance. By adding the notation for managing variability to the containment tree, the agility to instantiate variability from the very beginning is improved.

In addition, the top node component of the containment tree provides us with a decision model to make decisions about general user requirements. This decision model is used as proposed by KobrA, using the new containment tree to solve the logical and physical configuration of each new product during the application engineering phase. In this way, developers are able to generate new products by using the enhanced containment tree as the graphical interface to interact with the SPL framework. Moreover, domain engineers have the opportunity of dealing directly with the containment tree to update and maintain variability in the SPL framework in an agile way, making the work done during the domain engineering phase significantly more profitable.

As future work, we consider that it is necessary to work with the variability for each of our own developed components the KobrA way. Also, it is important to define how to deal with the open GIS components modified by SIGTel and how to integrate them in our architecture. In the future, our plans include to study how to use and exploit model-to-text transformations existing in Eclipse (M2T, [9]) in order to automate of translation process (i.e. transformation from logical components to code components). The first steps towards the SIGTel's SPL have been taken, but there is still a lot of hard work to be done.

Acknowledgments. This work has been partially sponsored by SIGTel Geomática, the EOGIS group (UCLM) and the Spanish Ministry of Science and Innovation through the Torres Quevedo Program and the DESACO (TIN2008-06596-C02-01) and AT (CONSOLIDER CSD2007-0022) Projects.

References

1. America, P., Obbink, H., Ommering, R., Linden, F.: CoPaM: A Component-Oriented Platform Architecting Method Family for Product Family Engineering. SPLC (2000)
2. Atkinson, C., Bostan, P., Brenner, D., Falcone, G., Gutheil, M., Hummel, O., Juhasz, M., Stoll, D.: Modeling Components and Component-Based Systems in KobrA. In: Rausch, A., Reussner, R., Mirandola, R., Plášil, F., et al. (eds.) The Common Component Modeling Example. LNCS, vol. 5153, pp. 54–84. Springer, Heidelberg (2008)
3. Atkinson, C., Bayer, J., Bunse, C., Kamsties, E., Laitenberger, O., Laqua, R., et al.: Component-based Product Line Engineering with UML, 1st edn. Addison Wesley (2001)
4. Dikel, D., Kane, D., Ornburn, S., Loftus, W., Wilson, J.: Applying software product-line architecture. Computer 30, 49–55 (1997)
5. Google, Google Maps (n.d.), `http://maps.google.com/`
6. Graphical Modeling Framework, `http://www.eclipse.org/modeling/gmp/?project=gmf-tooling#gmf-tooling`
7. Greenfield, J., Short, K., Cook, S., Kent, S.: Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools, Anaheim, California, USA (2003)
8. Kang, K., Cohen, S., Hess, J., Nowak, W., Peterson, S.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. SW Eng. Institute (1990)
9. Model to Text (M2T) project, `http://www.eclipse.org/modeling/m2t/` (last access July 08, 2011)
10. Moreno-Rivera, J.M., Osann, A., Calera, A.: SPIDER - An Open GIS application use case. In: First Open Source GIS UK Conference, Nottingham (2009)
11. Moreno-Rivera, J.M., Navarro, E.M.: Improving Component-based WebGIS Development by Adopting a Software Product Line approach. In: Information Systems Development, Praga, República Checa (2010)
12. Moreno-Rivera, J.M., Navarro, E.: Evaluation of SPL approaches for WebGIS development: SIGTel, a case study. In: 44th Hawaii Int. Conf. on Systems Science (HICSS-44 2011), January 4-7, pp. 1–10. IEEE Computer Society, Koloa (2011)
13. Pohl, K., Böckle, G., Linden, F.J.V.D.: Software Product Line Engineering: Foundations, Principles and Techniques, 1st edn. Springer, Heidelberg (2005)
14. MOSKitt, Home Page (n.d.), `http://www.moskitt.org/eng/moskitt/`
15. OGC Consortium, OpenGIS Implementation Specification for Geographic information - Simple feature access (n.d.)
16. OMG Common Variability Language (CLV), `http://www.omgwiki.org/variability/doku.php` (last access July 05, 2011)
17. OMG CVL Tool from SINTEF, `http://www.omgwiki.org/variability/doku.php?id=cvl_tool_from_sintef` (last access July 08, 2011)
18. SIGTel Geomática, `http://www.sigtel.es/` (last access June 27, 2011)
19. TOPCASED - Home, `http://www.topcased.org/` (last access June 27, 2011)
20. VarMod-Editor Web Page, `http://www.sse.uni-essen.de/wms/en/index.php?go=256#zi`

Towards a Development Framework Based on Behavior Recombination

Houman Younessi[1], Renaud Pawlak[2], and Carlos E. Cuesta[3]

[1] Rensselaer's Hartford, RPI
275 Windsor St, Hartford, CT 06120, USA
[2] LISITE – ISEP
28 rue Notre-Dame des Champs, 75006, Paris, France
[3] DLSI II, Rey Juan Carlos University
ETS Ingenieria Informatica, Mostoles 28933, Madrid, Spain

Abstract. This paper introduces a development framework that constructs software by recombining the behavior rather than the structure of the system. It would thus be possible to develop trusted scenarios that we call behavior sequences. System behavior then may be changed by changing some scenarios. We believe that our framework allows more confidence in evolution of software because the behavior of the system is entirely specified and tested upfront in the development process. Moreover, adaptation schemes are flexible since they rely on behaviors rather than components, which is the case in many approaches for software adaptation, which often use a structural perspective.

1 Introduction

Adapting software to a new context usually supposes execution variation points where decisions are made whether to run one strategy or another. For instance, if the context shows low power, we may want to adopt a strategy that consumes less, even if such strategy negatively impacts the QoS. Modular design is important for engineering software that must adapt, additionally we must assign each strategy to a well-defined module or service, with well-defined boundaries within the program. Thus, the conditional use of a module is a variation point and it becomes simple to activate or deactivate a given strategy at design- or at run-time, depending on the type of adaptation needed. An appropriate architecture makes plugging in or out a given module easy, enhancing loose-coupling, such as in component-based [12] and aspect-oriented architectures [13]. The main problem with most current approaches is that they rely on architectures that are imposed by choices made by designers. Such choices force modules to be defined, and the variation points can only be defined with regards to these modules. This problem has been acknowledged to be a limitation to the aspect-oriented approaches and has been referred to as the tyranny of the dominant dimension [8]. In general this is linked to architecture erosion [14], which occurs when software evolution is needed and when unanticipated variation points need to be added to the system. To overcome these issues, we have investigated an approach that consists of defining the variation points with regards to the behavior rather than the structure of the program. We propose a development framework that can be used to define trusted behaviors adaptions

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 632–636, 2011.
© Springer-Verlag Berlin Heidelberg 2011

of these behaviors. This framework can be explained as a rewriting rule-based language that allows for temporal logic expressions in the predicates of the rules themselves. The rewriting rules, will adapt the behavior of the system depending on conditions upon the behavior itself. With this approach, the structure of the program is not as important anymore, so it becomes possible to add unexpected variation points when some change occurs within the functional or environmental execution context.

2 The Behavioral Recombination Framework

In our framework, we see program behavior as a sequence of actions (state changes or messages) that occur during execution. Hence, adapting the program consists of changing the order of execution, or adding new states, to the initial sequence. Therefore, we have called this sort of adaptation *recombination* in analogy to recombinant DNA. The Behavioral Recombination Framework (BRF) consists of first modeling the behavior of the system with a behavioral recombination tool (a recombiner), of which we have developed a prototype [9]. With our tool, one can define trusted behaviors through the following actions:

1. defining the primary behavioral sequences of the system (i.e. the default behavior, when everything goes as expected),
2. defining rules that check that the primary behaviors work as expected,
3. defining rewrite rules that add alternate behaviors for deviation from the primary behavior e.g. alternate sequences, exceptions, errors, etc.

Such modeling is closely linked to the early development stages of many software engineering methods, especially the ones that promote definition of Use Cases [7]. In particular, the primary behavioral sequences can be seen as the main scenarios of Use Cases, and the rewrite rules as alternate scenarios. We believe that the Use Case scenarios to model is reasonable as it is a practical way to ensure what is being developed actually corresponds to what is expected. This is a belief shared by the agile development community that promotes Test Driven Development (TDD), and Continuous Integration [1]. In TDD, most integration tests are related to Use Cases and the tests actually play the role of programatic specifications, which can be automatically verified with a Testing Framework [4]. With our approach, we allow the specification of rules that can be seen as integration tests. As an example, imagine a usecase of a client that needs to connect to a server to upload a file. Within our framework, this can be defined with a simple scenario (a sequence of actions):

$$\begin{aligned} zip(file)\ ; \\ c = connect(server, user, password)\ ; \\ result = send(c, file)\ ; \\ disconnect(c) \end{aligned}$$

One of the implicit requirements is to save communication time. To do so, we decide to zip the file before uploading it to the server.

2.1 Behavior Predicates

Our framework relies on a language to express behavior predicates, that is to say expressions that can match sequences of actions. Conceptually, this predicate expression language is close to Linear Temporal Logic (LTL) [10], which allows evaluation of predicates on a sequence of program states. However, LTL is not always easy to use. In order to more easily express our predicates, we have defined a language closer to regular expressions. For instance $A* \, ; B$ will match sequences where A is verified any time and is followed by an action where B is verified (; stands for sequence operator, $*$ and $+$ for repetition, as in regular expressions). A and B are first order logic predicates on invocation and states. Both LTL and our language can be translated to Büchi automata [2][11] and are equivalent.

Besides writing the default behaviors, a first step within software construction could be to ensure that all the possible execution paths of the specification match some behavior predicates. For instance, we can write a predicate that ensures that zip is always called before send. This can be done with the following rule:

$$(\neg(zip(f) \vee send(?, f)))* \, ; send(?, f) \rightarrow raise\ error.$$

This rule can also be seen as a sort of integration test. It ensures that whatever the behavior of the system, it will raise an error when a file is sent without being zipped first. Hence, our development framework allows a TDD style. Behaviors can be adapted and refined with confidence since rules will raise error if a regression is made by mistake w.r.t. the expected behavior of the system.

2.2 Abstracting the Behavior

The rule defined in the previous section uses invocations to ensure correct system behavior. Using invocations needs that predicates know the underlying structure of the system, that is to say of the function prototypes that are used to implement behavior. If this structure changes during software construction, the rules may be broken, which makes the development process difficult to use. In order to avoid this issue, it is recommended to abstract the behavior in program states that can be defined through variables with predictive values. For instance, we could define a $zipped$ and an $upload$ variable to specify the states. This will give us the following behavior:

$$zip(file) \, ; \tag{1}$$

$$zipped = true \, ; \tag{2}$$

$$uploading = true \, ; \tag{3}$$

$$c = connect(server, user, password) \, ; \tag{4}$$

$$result = send(c, file) \, ; \tag{5}$$

$$disconnect(c) \, ; \tag{6}$$

$$uploading = false \tag{7}$$

We can thus rewrite the rule as: $uploading \wedge \neg zipped \rightarrow raise\ error.$

2.3 Recombining the Behaviors

Finally, our development framework allows for the definition of more complex behaviors with the use of rewrite rules that impact the original behavior. In a UseCase driven development process, these rewrite rules can be seen as alternate scenarios. Here, the following rule will retry the upload of the file if it fails (up to 3 times). It does so by installing alternate execution paths.

$count = 0$
$upload+ \ @ \ ; \neg upload \rightarrow$ // '@' is where the alternate path branches
  $count < 3 \wedge \neg result :$ // condition of the alternate path
    $uploading = true;$
    $count = count + 1;$
    $sequence(0);$ // insert the first subsequence matched by this rule
    $join$ // back to the primary path

The installation of this retry policy is done automatically by the behavioral recombiner, which builds up the entire behavioral model of the system. It uses the following algorithm:

- for all the sequences of the model, place the cursor on the first action,
- while the predicate of one rule matches the current action, apply that rule's action and calculate the new primary sequences (if rewrite rule),
- once all the rules have been applied, move the cursor to the next action.

At the end of the process, the recombiner has calculated the entire program behavior under the behavioral constraints defined by the behavior predicates (such as the ones in Sections 2.1 and 2.2).

3 Related Work

Our approach can be regarded as a form of Model Checking [3]. Like model checking, it calculates the entire state space of the application and can use temporal logic to validate properties on the program. For instance Spin [6] uses LTL formulas, which are very similar to our behavior predicates. One advantage of Spin, however, is that it supports multithreaded models, which is not (yet) the case with our approach. On the other hand, Spin does not support rewrite rules to construct the model. Another related work is Event-based AOP (EAOP) [5], which uses traces of program actions to trigger an advice. Event-driven aspects may be seen as closely related to our behavior rewrite rules. However, our approach is more a development framework used for specification and modeling, rather than a programming language like EAOP.

4 Conclusion

We propose a framework for constructing applications that is based on recombining sequences of actions so that the applications' behaviors are entirely specified and tested upfront. With this framework, currently under development, it will possible to develop trusted behaviors, and trusted adaptations of these behaviors, which is quite different from component-based adaptation strategies – yet both approaches may be complementary. We have already developed a simple recombiner in Java that enables the recombination of generic sequences, the basis for which has been explained in [9]. Since then, we have been working on generalizing the idea into our framework and integrate recombinant tools into a a closely compatible high-level requirements specification method, called Manhattan. We plan to extend our formalism to cover multithreading and validate it on simple but practical developments.

References

1. Beck, K.: Test-driven Development by Example. Addison-Wesley (2003)
2. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Proceedings of the International Congress on Logic, Method, and Philosophy of Science, pp. 1–12. Stanford University Press (1960)
3. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems 8(2) (1986)
4. Crispin, L., Gregory, J.: Agile Testing: A Practical Guide for Testers and Agile Teams. Addison-Wesley (2009)
5. Douence, R., Motelet, O., Südholt, M.: A formal definition of crosscuts. In: Proceedings of the 3rd International Conference on Reflection and Crosscutting Concerns. LNCS (2001)
6. Holzmann, G.J.: The model checker SPIN. IEEE Transactions on Software Engineering 23, 279–295 (1997)
7. Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G.: Object-Oriented Software Engineering - A Use Case Driven Approach. Addison-Wesley (1992)
8. Ossher, H., Tarr, P.: Multi-dimensional separation of concerns in hyperspace. Technical Report Research Paper RC21452, IBM Thomas J. Watson Research Center; Yorktown Heights; New York (April 1999)
9. Pawlak, R., Cuesta, C.E., Younessi, H.: Recombinant programming. Technical Report Research Report; Report No: 5380, INRIA (May 2006)
10. Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS), pp. 46–57 (1977)
11. Somenzi, F., Bloem, R.: Efficient Büchi Automata from LTL Formulae. In: Emerson, E.A., Sistla, A.P. (eds.) CAV 2000. LNCS, vol. 1855, pp. 247–263. Springer, Heidelberg (2000)
12. Szyperski, C.: Component Software: Beyond Object-Oriented Programming, 2nd edn. Addison-Wesley Professional, Boston (2002)
13. Tekinerdogan, B., Aksit, M.: Classifying and evaluating architecture design methods. In: Aksit, M. (ed.) Software Architectures and Component Technology: The State of the Art in Research and Practice, pp. 3–27. Kluwer Academic Publishers (October 2001)
14. van Gurp, J., Bosch, J.: Design erosion: Problems and causes. Journal of Systems and Software 61(2), 105–119 (2002)

Dynamic Adaptation through Event Reconfiguration

Truong Giang Le, Olivier Hermant,
Matthieu Manceny, and Renaud Pawlak

LISITE - ISEP, 28 rue Notre-Dame des Champs,
75006 Paris, France
{le-truong.giang,olivier.hermant,
matthieu.manceny,renaud.pawlak}@isep.fr

Abstract. We introduce a new programming language called INI, which eases the development of self-adaptive software. INI combines both rule-based and event-based programming paradigms, by allowing the definitions of rules that can be triggered by events. Besides a convenient language-level support for synchronization, the key point with INI events is that they come with a configuration layer, which is set up through input parameters passed to the events when created. Additionally, events can be stopped, reconfigured, and restarted at runtime by the program itself. Ultimately, this re-configuration can be triggered by other events, thus allowing the program to adapt to new operational environments.

Keywords: Adaptive systems, rule-based programming, event-based programming, reconfiguration.

1 Introduction

The role of software in our society has become more and more important and to satisfy higher demands from customers, software systems must become more "versatile, flexible, resilient, dependable, robust, energy-efficient, recoverable, customizable, configurable, and self-optimizing" [1]. However, developing and maintaining software systems is hard and resource-consuming because they are operating in an environment that is not well defined or predictable. Consequently, software systems need a new and innovative approach for evolving and running. Currently, one of the most promising approaches to achieve such properties is building self-adaptation and self-management mechanisms [8].

In recent years, with the development of autonomic computing, software engineering researchers have been motivated to pay more attention to design and develop self-adaptive software systems. Basically, adaptive software comes with the assumption of dynamic environment while non-adaptive software does not [14]. By using rules explicitly or implicitly, most self-adaptive software can choose reactions to monitored events happening during execution [13]. Therefore, one of the current research trends is to develop a programming language to easily

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 637–646, 2011.
© Springer-Verlag Berlin Heidelberg 2011

express and capture those changes so that the systems can modify their behavior at runtime. In this paper, we introduce a new programming language called INI, which we designed to assess rule-based and event-based oriented programming. In INI, we allow programmers to define events, which need to be captured and monitored. Events are usually executed asynchronously but can synchronize on other events with intuitive language constructs. Moreover, events can be reconfigured at runtime through a well-defined reconfiguration mechanism. As a result, when something happens during execution, INI program can capture and adapt to these changes automatically by reconfiguring its events.

The rest of this paper is organized as follows. In Section 2, we give an overview of related work. In Section 3, we discuss events in INI, including its syntax, built-in events and user-defined events. Events synchronization and reconfiguration are shown in Section 4. In that section, we also demonstrate the capabilities of INI in handling variabilities in the operational environment with a small example. Finally, some conclusions and future work are presented in Section 5.

2 Related Work

A general overview of research and problems on self-adaptive software can be found in [7,12]. The key aspect of self-adaptive software is that code behavior is evaluated or tested at runtime, which may lead to a run-time change in behavior [7]. As a result, the run-time code should contain the following mechanisms:

- A mechanism to detect changes during execution
- A mechanism to switch algorithms and operations due to these above changes

Laddaga [7] discussed two useful paradigms for utilizing self-adaptive software. One of these is the planning paradigm, and the other is the control paradigm. In [13], Wang proposed a Rule Model for self-adaptive software, which includes three key concepts (event, parameter, and rule). In Wang's paper, rules were used to adjust software's behavior at runtime. Psaier *et al.* [11] presented a middleware for programming and adapting complex service-oriented systems. Their approach was based on monitoring and real-time intervention to regulate interactions based on behavior policies.

Cheng *et al.* [2] introduced a new language of adaptation, which ultimate aim is to automate human tasks in system management to achieve high-level stakeholder objectives. They claimed that such a language is "expressive enough to describe human expertise and comes with the flexibility and robustness to capture complex and potentially dynamic preferences".

In our work, we develop INI by combining both rule-based and event-based programming styles. INI takes inspirations from existing rule-based languages [3], and event-based programming similarly to the work above. With rule-based programming, an action is executed only if its guard is evaluated to true in the context of the function. With event-based programming, we allow programmers to write their own events to capture and monitor changes happening during execution. Moreover, we fully support events synchronization and reconfiguration. Although several event-based programming languages have been proposed so far [4,6], their support for writing self-adaptive software are limited. To our knowledge, no languages support event reconfiguration at runtime. The key feature of INI compared to all the languages we have studied is that events are dynamically reconfigurable. This feature allows INI programs to automatically reconfigure themselves to adapt to changes in the environment.

3 Programming with INI

The interpreter of INI is currently implemented in Java and runs on a JVM, but INI's syntax and semantics are not Java's ones. Each INI program contains functions, in which the starting point is the function `main`. All functions are written with rules in combination with event expressions. Each rule is defined explicitly with a guard and an action. The guard triggers the evaluation of the action when evaluated to true in the context of the function. Along with rules, programmers can use events in INI to capture changes when their programs are running. In this paper, we focus on event-triggered rules since we use them as the main mechanism for self-adaptation.

3.1 Events in INI

Table 1. Some built-in events in INI

Built-in event kind	**Meaning**
`@init()`	The `@init` event is invoked when the function starts evaluating.
`@end()`	On the contrary to the `@init` event, the `@end` event is triggered when no more event can be applied and when the function is about to return.
`@every[time:Integer]()`	The `@every` event occurs periodically as specified by its input parameter (in milliseconds).
`@update[variable:T] (oldValue:T,newValue:T)`	The `@update` event is invoked when a specific variable is changed during execution.

An event is defined as something happening during the execution of the program, which should be monitored and handled. Events in INI can be understood at three different levels: the *event kind* corresponds to the class/type of the event; the *event instance* (*event* for short) corresponds to the definition of an event-triggered rule in the INI program; the *event thread* corresponds to the execution of an event instance. By convention, event kinds start with `@` and can take input and output parameters. Input parameters can be understood as configuration parameters to tune the event execution. Output parameters are passed to the action. These parameters may be optional. Syntactically, each event (instance) is written as below. Optionally, an event can also be bound to an id, so that other parts of the program can refer to it.

```
id:@eventKind[inputParam1=value1, inputParam2=value2, ...]
             (outputParam1, outputParam2, ...) { <action> }
```

To allow programmers to write code more easily and conveniently, INI comes with some common built-in event kinds such as `@init`, `@end`, `@every`, `@update`. The meanings of these events are given in Table 1. For example, the following code creates an `@every` instance called `e`, which increments `v` every second. The `@update` instance `u` triggers the action (event handler) that prints out the `v` variable value when it changes, i.e. every second.

```
e: @every[time=1000]() { v = v + 1 }
u: @update[variable=v](oldv, newv) { println("v=" + newv) }
```

Beyond built-in events, we also allow programmers to write their own event kinds in Java or in C and integrate them into an INI program as explained in the next section.

3.2 Example

In this section, we discuss a sample INI program, written in an event-driven style. In our example, we use a video camera to detect the movement of a ball, get its positions in space periodically, and save them into a CSV file.

To do so, we first need to define a new event kind to detect the ball and send its position to the program when detected. Our event has one input parameter called `frequency`. This parameter is applied to set how long the event should sleep between two image detections (time unit is in milliseconds). Besides, we have three output parameters `(r,x,y)`, which are the radius and coordinates of the detected ball in the captured image. In Java, we need to subclass the `ini.event.Event` class to define the behavior of our new event kind as shown in the code below.

```
public class BallDetection extends ini.event.Event {
  Thread ballDetectionThread;
  @Override public void eval(final IniEval eval) {
    (ballDetectionThread = new Thread() {
      @Override public void run() {
        ...
        do {
          try {
            // Sleep as long as the configuration indicates
            sleep(getInContext().get("frequency")
              .getNumber().longValue());
            // Use OpenCV to detect the ball
              ...
            // Write data to output parameters
            variables.put(outParameters.get(0), r);
            variables.put(outParameters.get(1), x);
            variables.put(outParameters.get(2), y);
            // Execute the event action
            execute(eval,variables);
          } catch (Exception e) {...}
        } while (!checkTerminated());
```

```
22        }
23      }).start();
24    }
25    @Override public void terminate() {...}
26  }
```

In the BallDetection class, the method eval will be upcalled by the INI evaluator when the program uses our event. It creates a thread that sleeps accordingly to the event configuration (line 10), detects the ball using the OpenCV library (line 13) [5] and write the results in output parameters to be passed to the INI program (lines 15-17). The event-triggered action, also called *event thread*, is executed at line 19 using the execute method provided by the API, which by default runs asynchronously. Finally, the method terminate is overridden to stop the event: INI upcalls this method when the program exits or forces the event to shutdown. To understand more about built-in events in INI and how to write user-defined events, interested readers may refer to INI Language Reference Documentation [10].

```
1  import "lib_io.ini"
2  @ballDetection[frequency:Integer](Float, Integer, Integer)
3    => "ini.ext.events.BallDetection"
4  function main() {
5    @init() { f = file("ballData.csv") }
6    // Use our event get notified for ball detection
7    @ballDetection[frequency = 1](r,x,y){
8      case {
9        !file_exists(f) {
10         // Create a new CSV file to store data
11         create_file(f)
12         fprintln(f, "Time,␣r,␣x,␣y")
13       }
14     }
15     fprintln(f,to_string(time())+","+r+","+x+","+y)
16   }
17 }
```

Fig. 1. A sample INI program with event-driven style

In Figure 1, we write the actual INI program, which binds our Java user-defined to the @ballDetection event kind (lines 2-3). In the @init event, we define a variable f, which indicates the CSV file we want to store data in after collecting the ball positions over time. In INI, all variables are global within the function where they are defined. This means that f can be accessed anywhere in the function main. In our program, the @ballDetection event is triggered periodically, i.e. each millisecond. If a ball is detected, we check whether the CSV file exists or not (line 9). If the file does not exist, we create a new one and write the header. Finally, we write the data to the file (line 15), including the time when the ball was detected and its position.

4 Events Synchronization and Reconfiguration

The event reconfiguration mechanism is developed in INI so that we can modify the input parameters (configuration) of the event-triggered rules at runtime. For example, an event can be notified for a change in the environment and decide to reconfigure another event to adapt the program to the new environment. However, to achieve safe reconfiguration, synchronization among events is also essential. In this section, we first explain event synchronization, and then discuss event reconfiguration, which we illustrate using our ball-detection example.

4.1 Events Synchronization

Except for the `@init` and `@end` events, most INI events are executed asynchronously by default. From an implementation perspective, it means that each event handler runs in its own thread, potentially simultaneously to other events. In many cases, a given event `e0` may want to synchronize to other events `e1, ..., eN`, which means that the synchronizing event `e0` must wait for all the target event threads to be terminated before running. For instance, it can be used to ensure that a shared variable accessed both for reading and writing has a consistent state, or also to ensure that all event threads have terminated before reconfiguring the associated event-triggered rule. Thus, INI provides a specific language construct: the symbol `$` along with the list of identifiers of target events with which we want to synchronize. For example, the following code ensures that the above discussed event `e0` synchronizes `N` target events.

```
$(e1,e2,...,eN) e0:@eventKind[...](...) { <action> }
```

Note that one of the target events can also be synchronized with `e0`. Cross-synchronization of events means that their executions are mutually exclusive. We now precisely define the semantics of synchronization in INI.

To implement synchronization in INI, we use one lock and one *count* variable associated with each event. The lock is an instance of `java.util.concurrent.locks.ReentrantLock` and is used to avoid concurrent execution when required. We use the functions `lock` (blocking), `tryLock` (non-blocking and returns true of false depending on whether the locking was successful or not), and `unlock`. For more details, refer to the Java document of the methods of the same names in the `ReentrantLock` class [9]. The *count* variable holds the number of threads currently executing for the associated event. We use the following notation: for an event e_i, we call l_i its associated lock and $count_i$ its associated thread counting variable.

Let $(e_1, e_2, ..., e_N)$ be the list of target events with which e_0 synchronizes. To execute the event e_0 in INI, we apply Algorithm 1, which also applies to any event execution in the INI system. We can see that the execution of events

Algorithm 1 Our algorithm to execute e_0 synchronized with $(e_1, e_2, ...e_n)$

```
1: repeat
2:    allLocked := true
3:    lock(l_0)
4:    for i := 1 to N step 1 do
5:       if ¬ tryLock(l_i) then
6:          allLocked := false
7:          for i := 1 to i − 1 step 1 do
8:             unlock(l_j)
9:          end for
10:         unlock(l_0)
11:         sleep(randomTime())
12:         break
13:      end if
14:   end for
15: until ¬allLocked
16: for i := 1 to N step 1 do
17:    wait-until count_i = 0
18: end for
19: count_0 := count_0 + 1
20: for i := 1 to N step 1 do
21:    unlock(l_i)
22: end for
23: unlock(l_0)
24: eval(e_0)
25: count_0 := count_0 − 1
```

includes four steps. First, e_0 locks its own lock and tries to lock all target events (lines 1-15). When all events are locked, the event e_0 needs to wait until all other events are terminated (lines 16-18). The `wait-until` mechanism in our algorithm is currently implemented with Java monitors and thread notification. Next, the number of threads executing for event e_0 is incremented and locks for all events are released (lines 19-23). Finally, the event can be actually evaluated and when it is terminated, the number of running threads for it is decremented (lines 24-25).

4.2 Example Using Synchronization

Let us take again our example shown in Figure 1. Assuming that after we collect data about positions of a ball in space, we need to upload the CSV file to an FTP server. To do this, we create a new `@every` event as shown in Figure 2. Inside this event, first, we compress data and then upload the compressed file. Since we collect data periodically, we use a timestamp (line 4) to distinguish data at different time.

```
1 ...
2 // Compress and upload the collected data to an FTP server
3 @every[time = 5000](){
4   t = time()
5   zip(file_name(f),to_string(t)+"ballData.zip")
6   upload_ftp("ftp_address", "user_name", "password",
7     to_string(t)+"ballData.zip",
8     to_string(t)+"uploadedBallData.zip")
9 }
```

Fig. 2. An event used to upload collected data

By default, the two events `@ballDetection` and `@every` will be executed asynchronously. However, it is essential that when we collect data (`@ballDetection`), the uploading process (`@every`) should not happen and vice versa. In other words, these two events need to be mutually exclusive. To ensure this, the programmer needs to modify the program at lines 7 of Figure 1 and 3 of Figure 2 by naming the two events and use their identifiers to cross-synchronize them, as it is shown in the following code:

```
$(e) b:@ballDetection[frequency = 1](r,x,y) { ...
$(b) e:@every[time = 5000](){ ...
```

4.3 Event Reconfiguration

In this section, we will demonstrate the capabilities of INI in handling changes happening in the environment though the event-reconfiguration mechanism. Event reconfiguration consists of changing the values of the event-triggered rule input parameters. Programmers can call the built-in function `reconfigure(eventId, [inputParam1=value1, inputParam2=value2,...])` to reconfigure their events. Moreover, we also allow programmers to stop and restart events with the built-in functions `stop_event(eventId)` and `restart_event(eventId)`. Typically, it is required to stop an event before reconfiguring it.

Let us now consider that our ball-detection example of Section 3 runs in an embedded environment where the power is supplied by a battery. One way to take into account this new constraint is to adapt the data-collection frequency to the power level. First, we add a new user-defined event kind called `@powerAlarm`, which notifies the program each time the power level passes a given threshold both ways, when charging or discharging. This event has one input parameter named `threshold`, and one output parameter named `currentLevel`, which tells us if the level is currently lower or greater than the threshold. When the program below is running, if it detects that the power-level is lower than 50% (configured line 1), it stops the event `b:@ballDetection` (line 4), then changes the value of its input parameter (i.e., `frequency`) to 1000 (line 5), and finally restarts it (line 6). Conversely, if the power goes over the threshold the value for the parameter `frequency` is set again to 1 (lines 8 - 10). The event `@powerAlarm` is synchronized with events `b:@ballDetection` and `e:@every` as specified by `$(b,e)` at line 1 in order to avoid unfinished detection or upload jobs to be stopped.

```
1  ...
2  // Adapt the ball detection frequency at the 50% threshold
3  $(b,e) p:@powerAlarm[threshold = 0.5](currentLevel) {
4    case {
5      currentLevel < threshold {
6        stop_event(b)
7        reconfigure(b, [frequency = 1000])
8        restart_event(b)
9      } default {
10       stop_event(b)
11       reconfigure(b, [frequency = 1])
12       restart_event(b)
13     }
14   }
15 }
```

Finally in the following code, we show how to add a second `@powerAlarm` instance that stops all other events when the threshold goes below 10%.

```
$(b,e) @powerAlarm[threshold = 0.1](currentLevel) {
  case {
    currentLevel < threshold { stop_events([b, e, p]) }
    default { restart_events([b, e, p]) }
  }
}
```

5 Conclusion and Future Work

In this paper, we introduce a new language called INI, which can be used to write self-adaptive software through events synchronization and reconfiguration mechanisms. Programs written in INI may dynamically change their behavior or improve their operation depending on dynamic operational conditions of the environment. With the example that we used along the paper to demonstrate the capabilities of INI, we can see that using INI to write new event-driven rules does not require too much effort. The programmers can capture the changes during execution by writing their own user-defined events and then define suitable actions for them. The changes necessary to take into account new constraints can thus be localized and encapsulated within some rules, which implies code maintainability and robustness with regard to changes.

For future work, we will improve INI so that it can handle changes in environment more efficiently. For example, as stated in [2], we need a mechanism to handle failure during adaptation execution. If a failure happens when our systems try to adapt to the environment, it is essential to recover from that failure. Besides, we will define formal semantics for validating and verifying useful properties of INI programs and prove the soundness of its type system. Finally, we also have a plan to perform more case studies to evaluate the capabilities of INI.

References

1. Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.): Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525. Springer, Heidelberg (2009)
2. Cheng, S.W., Garlan, D., Schmerl, B.: Architecture-based self-adaptation in the presence of multiple objectives. In: Proceedings of the 2006 International Workshop on Self-Adaptation and Self-managing Systems, SEAMS 2006, pp. 2–8. ACM, New York (2006)
3. Cirstea, H., Moreau, P.-E., Reilles, A.: TomML: A Rule Language for Structured Data. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 262–271. Springer, Heidelberg (2009)
4. Fischer, J., Majumdar, R., Millstein, T.: Tasks: language support for event-driven programming. In: Proceedings of the 2007 ACM SIGPLAN Symposium on Partial Evaluation and Semantics-based Program Manipulation, PEPM 2007, pp. 134–143. ACM, New York (2007)
5. Garage, W.: OpenCV, http://opencv.willowgarage.com/wiki/
6. Kamina, T., Aotani, T., Masuhara, H.: Eventcj: a context-oriented programming language with declarative event-based context transition. In: Proceedings of the Tenth International Conference on Aspect-oriented Software Development, AOSD 2011, pp. 253–264. ACM, New York (2011)
7. Laddaga, R.: Self adaptive software problems and projects. In: Proceedings of the Second International IEEE Workshop on Software Evolvability, pp. 3–10. IEEE Computer Society, Washington, DC (2006)
8. Müller, H.A.: Towards self-adaptive software-intensive systems. In: Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution (IWPSE) and Software Evolution (Evol) Workshops, IWPSE-Evol 2009, pp. 7–8. ACM, New York (2009)
9. Oracle: Java SE 6 Documentation (June 2011), http://download.oracle.com/javase/6/docs/
10. Pawlak, R.: INI Online (June 2011), http://perso.isep.fr/rpawlak/ini/
11. Psaier, H., Skopik, F., Schall, D., Juszczyk, L., Treiber, M., Dustdar, S.: A programming model for self-adaptive open enterprise systems. In: Proceedings of the 5th International Workshop on Middleware for Service Oriented Computing, MW4SOC 2010, pp. 27–32. ACM, New York (2010)
12. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. ACM Trans. Auton. Adapt. Syst. 4, 14:1–14:42 (2009)
13. Wang, Q.: Towards a rule model for self-adaptive software. SIGSOFT Softw. Eng. Notes 30, 8–12 (2005)
14. Yoo, C., Jung, W., Park, D., Lee, B., Kim, H., Wu, C.: An adaptive software framework based on service composition. In: Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management & Applications, SERA 2007, pp. 476–484. IEEE Computer Society, Washington, DC (2007)

Towards a Reference Architecture for Mashups

Tommi Mikkonen and Arto Salminen

Tampere University of Technology, Korkeakoulunkatu 1,
FI-33720 Tampere, Finland
{tommi.mikkonen,arto.salminen}@tut.fi

Abstract. Software mashups – web applications that combine content from multiple web sites to an integrated experience – have rapidly become a popular trend. Due to the novelty of mashup development, methods, tools, and architectures for creating mashups are still rather undeveloped, and there is little engineering support behind them, even if the most common architectural requirements for composing mashups can be easily identified. In this paper we present a list of architectural issues and derive a reference architecture to serve as a starting point for the design of new mashups. The long-term goal of our work is to facilitate the development and maintenance of robust, secure, and compelling mashups, and more generally ease the transition towards web-based software development.

Keywords: Web applications, mashups, web programming.

1 Introduction

The Web has been a success of almost an unbelievable scale. Within two decades the fashion information is distributed has changed radically, making the Web the first truly global information system where anything that is put online is available in global scale immediately. These abilities are changing numerous lines of business, such as entertainment and banking, and we expect that a number of other businesses – including software development – will follow.

The consequences of the above transition are many. The ability to download and combine code and content from multiple sources from anywhere in the world is opening up entirely new possibilities for software development. Web applications that rely on these capabilities, commonly referred to as mashups, have rapidly become common, and there are numerous compelling examples of the benefits of such applications. Furthermore, the trend towards software mashups has given rise to a number of environments and tools that have a specific objective to make mashup development easier. However, despite the recent research interest in mashups, their design is commonly considered an opportunistic, ad-hoc activity with minimal relation to software engineering practices, architecting, or disciplined development in general [4].

In this paper, we aim at making mashup development more disciplined by introducing architectural design drivers for mashup development and by deriving a reference architecture for composing mashup applications. The background of this paper is at implementing numerous systems combining data from the web and local

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 647–656, 2011.
© Springer-Verlag Berlin Heidelberg 2011

content [11, 7]. Some of the applications are running inside the web browser, whereas some other are built using more optimized approaches, with the main lessons learned documented in [12]. Experiences from these have given us a unique view to creating mashups in general and to programmatic development of client-side mashups in particular.

The structure of the paper is the following. In Section 2 we provide an overview to mashups in general. In Section 3, we discuss architectural requirements that are commonly identified in mashup development, and in Section 4 we introduce a reference architecture for composing mashup applications. In Section 5 we describe a mashup example implementing the reference architecture. In Section 6 we finally draw some final conclusions and identify directions for future work.

2 Mashup Applications and Mashup Development – An Overview

A typical software mashup combines data, images, code, and other content from multiple sources into a new user experience. In today's mashups data and content are downloaded most commonly from different web sites, but in principle data from any source (such as the user's local computer or an intranet database) can be used. In addition to aggregating data from multiple sources, mashup applications typically provide an alternative user interface or add advanced filtering or visualization capabilities to a web service.

As an example, Fig. 1 presents a mashup application that combines maps from Google Maps (http://code.google.com/apis/maps/) and road weather condition information from Finnish Road Administration (http://www.tiehallinto.fi). The general idea of the application is to use Google Maps API to determine an optimal route between two points chosen by the user on the map of Finland and display all weather cameras along the route as markers. When the user clicks on any of the markers on the map, a live image and current weather conditions form the selected camera are fetched and presented.

Both server-side and client-side mashups can be implemented. Today, most mashup development tools (see a summary of the tools in [14]) are intended for server-side use, that is, the downloading, processing, and generation of web content is performed on the server. Many of these server-side environments provide a hosting service from which the generated mashups cannot be removed. Another common theme is that the development of the mashups takes place in the web as well, with the browser acting as the user interface. Server side mashup tool architecture based on layers has been studied by López et al. [6]. Their architecture consists of four layers: *source access* (accesses web resources), *data mashup* (creates structural presentation of data), *widget* (holds all widgets available in the system) and *widget assembly* (creates an user interface) layer. In addition to the layers the architecture includes *common services* that provides general functionalities and can be used from any layer. The result mashup created with this architecture is similar to a web portal with the exception that the widgets are connected.

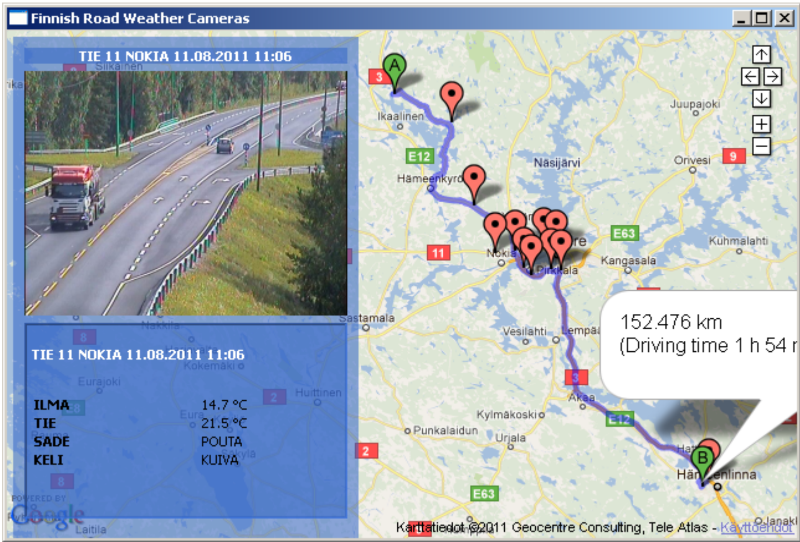


Fig. 1. Sample mashup application

In client-side mashups, in contrast, the downloading and combination of web content is performed on the client, most commonly in a web browser running on a desktop computer. A typical implementation is utilizing the JavaScript language [2] and additional JavaScript libraries, despite their well-known pitfalls [1]. This approach suffers from various technical limitations that arise from the design of the web browser. One example of such is the same-origin principle that has been designed to prevent a web client from downloading data from multiple web sites (https://developer.mozilla.org/en/Same_origin_policy_for_JavaScript). However, while the web browser is commonly used, also other kinds of applications are possible. In particular, when devices that are limited in their capabilities – such as mobile phones, house-hold appliances, or embedded systems – are used, other approaches can provide additional benefits in terms of performance, responsiveness, and memory consumption [8, 7].

It is important to make clear distinction between portals and mashups. Portals are web pages that contain information that is retrieved from different sources. Usually the user interface of a portal consists of numerous “portlets”, separate pieces of content or widgets that are shown up in unified way. However, in contrast to mashups, portlets are isolated into their own units and cannot communicate with each other. Mashups are more integrated applications that can be constructed so that the user cannot distinguish data origins based on its appearance in the user interface. Usually portals’ content is aggregated exclusively on the server-side, whereas mashups can be composed either server or client side. Furthermore, mashups create something new from the information they are based on, instead of just aggregating the data into a single view.

There are also some common elements in the development of mashups. For instance, common file formats such as XML and JSON, techniques such as RESTful web services [3], and JavaScript libraries including –to mention a few of the many – systems such as Scriptaculous (http://script.aculo.us/), jQuery (http://jquery.com/) and Dojo (http://www.dojotoolkit.org/) have turned out to be invaluable and repeatedly used in mashup development.

In general, mashup development has gained a lot of research interest recently. Different patterns and trends in mashup development can be identified. For example, Wong et al. have categorized mashups into five different groups: *aggregation*, *alternate UI & in-situ use*, *personalization*, and *focused view of data and real time monitoring* [17]. Another paper by Lee et al. presents seven mashup patterns: *data source*, *process*, *consumer*, *enterprise*, *client-side*, *server-side* and *developer assembly mashups* [5]. In addition, a number of challenges related to mashup development have been pointed out. As stated by Zang et al. [18], mashup developers encounter problems mainly in three areas: *API functionality*, *documentation* and *coding details*. Issues related to API functionality in their research were, for example, authentication and performance problems. Some developers were concerned about the lack of proper documentation at all levels, including API reference, tutorials and examples. The programming skills needed for creating compelling mashups in JavaScript were also identified as hard to learn. Finally, the relation of disciplined software engineering principles and mashup development – or, even more generally, the development of web applications – remains vague [9].

3 Architectural Requirements for Mashup Applications

Despite the popularity of mashups, the current view is that due to the relatively *ad hoc* nature of mashups, it has been – and still is – difficult to provide developers with general-purpose tools or uniform development guidelines for mashup development. While the development of mashups has sometimes been considered largely ad-hoc activity – see for instance [4] – there are commonalities in the goals of different mashup systems. To begin with, all these projects build on top of parallel activities and the innovativeness of a large group of independent service providers and aim at providing superior user experience without overlooking other important quality attributes of the system, such as security, performance, availability, and modifiability. Furthermore, since mashups in general are built by using the available means for combining components, mashup projects can build upon other mashups.

In the following, we address generic functions that are recurring themes in mashup development. The list by no means conclusive – for instance performance, scalability, and security issues must be taken into account separately since these often are cross-cutting themes in software design.

Service Interfacing. Mashups are generally very vulnerable to changes that occur when the data and interfaces offered by web services are modified. It is also relatively common for web services that are still under development or of beta quality to become unavailable for extensive periods of time. Mashups that rely on the availability of specific data and specific data formats and interfaces tend to break down easily when such changes occur. This particular property has been the most common problem when running mashups we have developed. Consequently, mashups must be architected such that it is easy to modify the interactions that take place between the core application and services the system uses. Another interesting challenge is the overlapping nature of web. It is common that there are several competing services providing similar functionality. For example, a map component, a commonly used item in mashups can be obtained from numerous sources, including for instance services such as Google Maps (http://maps.google.com/), Microsoft Bing (http://www.bing.com/maps/), Yahoo! Maps (http://maps.yahoo.com/) or OpenStreetMap map (http://www.openstreetmap.org/). To complicate the selection, the terms of use, availability of the services, features offered by the sites, and response times can vary considerably between different services. Moreover, the interfaces of these services are different, sometimes proprietary, and – for some of the services – relatively unstable and poorly documented. In order to avoid problems, the interfacing component must be such that it is easy to replace one service with another one that offers related service. Finally, there are also technical restrictions associated with interfacing. For instance, some web services have limitations in their APIs regarding the maximum number of requests that are allowed from a single domain within a certain period of time – such limitations are important in protecting the service from denial of service (DoS) attacks. Furthermore, it is common for service providers to utilize mechanisms that monitor the number of requests from a single domain. Some popular services such as Google Maps require a proper API key when making Maps API calls. An API key is a unique string that is issued by the service provider to identify application and bind it to a previously issued domain. Typically license terms strictly forbid the sharing of API keys. This can be a problem when developing widgets or independent web applications that do not use the typical browser-based approach to access the service.

Data handling, conversions, and exchange. Data handling, conversions and exchange is often a key function in mashups. Since mashup development by definition involves numerous web sites, it is inevitable that data formats can be different. Therefore, in order to create new, compelling applications that work with data, the data coming from different sources must almost always be transformed into a common format that preferably is easy for the application to process. This way application logic – the part that is responsible for the actual mashup function – can make minimal assumptions about what services are being used. This in turn adds an additional layer of flexibility between use and access of downloaded data. In today's mashups, data exchange is commonly based on established formats such as JSON and XML, but in many cases application-specific or service-specific formats must be used. At times, the developer even has to scrape and parse the required data manually from an HTML page using the limited facilities that are available in the web browser. This can be rather cumbersome and slow, not to mention the fact that even small changes in the page can be devastating for the application. Finally, the asynchronous loading of data that is characteristic of web applications today often causes timing problems when data is received from multiple sources.

Application logic. In the design of mashups, the actual application logic is constituted by the processing needed for generating the actual mashup. Therefore, this particular piece of the system is the holder of the majority of application logic. Depending on the tools that are being used, there are several ways to represent the logic. Some systems, such as Yahoo Pipes (http://pipes.yahoo.com/) allow one to graphically denote how processing proceeds. In contrast, if the whole system is composed using JavaScript, the logic is most commonly composed as a piece of executable program code. The mashup function can be produced by the end users, and no professional developers are necessarily needed [10, 16]. For obvious reasons, the different techniques are suited for different purposes, and no generic guidelines regarding what tool to use can be given.

User interaction. In order to create enhanced user experience, the users almost all need some sort of visual system to interact with. While sometimes the rendering is closely associated with one of the services – in particular, map-based applications are commonly such – rendering must be a configurable and modifiable subsystem. There are several motivations for separating user interactions into a separate module. For instance, different widget sets may be supported, some of which are intended for native use and some for being used inside the browser. Similarly, internationalization requirements may pose special needs that must be dealt with at this level. Moreover, in some cases the user interface can be displayed in a separate device, similarly to what can be achieved with consumer electronics using DLNA (http://www.dlna.org/) or UPnP (http://www.upnp.org/) technologies.

4 Deriving Reference Architecture

Despite the above common elements, little attention has been paid on how to take them into account in the architecture design. Instead, the creators of these projects have often come to their solutions intuitively or via trial-and-error. Hence there is little guidance for the creators of new mashup systems regarding to how to design the architecture that forms the basis of the architecture, even if the basis for rational design can be established.

Next, we describe the fundamental architectural structure for client-side mashup development (Fig. 2). As is common with reference architectures, practical implementations can be built where some of the components have been merged or omitted totally, if they play no role in the application that is being constructed.

Content providers. In order to extract content, there needs to be interfacing capabilities to the content providers, be it data in the web, local data residing in the device itself, or something that is generated on the fly depending on the context, such as GPS data. While not a part of the reference architecture per se, these are by definition an integral part of any mashup system, be it a system that is building on top of already existing Web services or simply something that is readily available in the device itself.

Content formatter and extractor. Content formatter and content extractor together compose the mashup data model. The data model is responsible for managing and maintaining the data that is used by the application for various purposes. In particular, the data model includes operations such as data accessing operations and extractors that download data from the web, from the local file system or from devices peripherals, as well as data formatters that manipulate the data provided by the extractors into a form that is easier to work with. Furthermore, the component may also maintain a local storage to support offline use of the service. Internally, content formatters and extractors can be layered, with different layers being responsible for different tasks, such as low-level interfacing, data scraping, formatting changes, and so forth.

Mashup creation. In order to combine data, there needs to be a mashup function that performs the actual mashup related operations. As is common with application logic, the fashion the actual operations are implemented is different for different applications. Some applications are composing content on the screen in different layers, placing for example graphical shapes on top of a map offered by one of the content providers, some other systems apply pipes-and-filters architectural style where content is processed in different ways, and so on.

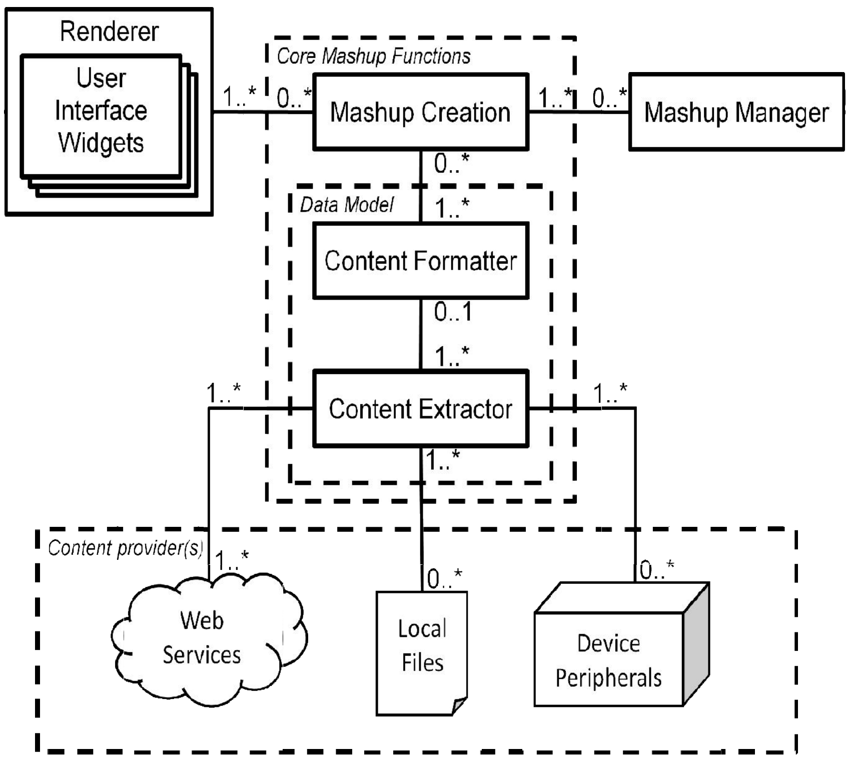


Fig. 2. Reference architecture for mashup applications

Renderer. Since mashups are commonly used to visualize available data it is obvious that a user interface layer is needed. Because visualization commonly needs graphical elements, most mashup systems rely on an existing set of user interface widgets that usually also support interaction with the human user. Numerous widget libraries readily exist, and the use of a certain set of widgets often has a major effect on the look and feel of the application. A further detail associated with widgets is that there are different binding techniques to connect widgets and data, which may have an effect in the design. Furthermore, if the operations are simple, or if the widget set has been designed so, it is sometimes possible – or even obligatory – to merge mashup related operations with the user interface. However, in the general case, these functions are distinct: the user interface is responsible for rendering the data produced by the mashup function and for interacting with the user in general, whereas the mashup function performs the actual mashupping that creates the data to be rendered. This is clearly visible in systems where rendering takes place in an external device like a TV set.

Mashup manager. To overcome the fragile nature of mashups and be able to make updates when necessary a mashup manager is introduced. This component takes care of ensuring that mashup content is available and has ability to adapt to changes that may occur for instance when mashup content providers change their interfaces. For example, the subsystem can decide to use a backup service when the primary service is failing. Similarly, the manager is responsible for defining whether the system should be run in online or offline mode, and for selecting the renderers that are to be used. The manager is also the main responsible component for issues such as security, which however can affect also other components.

5 Example Implementation

As an example of a mashup following the reference architecture, we discuss more about the application shown in Fig. 1. It uses Lively for Qt (http://lively.cs.tut.fi/qt) – a custom mobile runtime environment that makes the Qt APIs (http://qt.nokia.com/) accessible from JavaScript code [8].

The application consists of three JavaScript objects: *QtWeatherCameras* that implements most of the mashup creation functionality, *ImageViewer* that is used as a part of the user interface to display weather data and *ErrorManager* that implements mashup manager functionality. In the following correspondence between these parts of the application and the reference architecture are described in more detail.

QtWeatherCameras. QtWeatherCameras has functions to extract road weather data from the Finnish Road Administration, the map and optimized route from the Google Maps API and weather camera locations from a local file. It reacts to users actions and fetches the optimal route using Google Maps API. The fetched route is filtered (content formatting) and combined with information about weather camera locations (mashup creation). Finally the result is shown as markers on the map (rendering). When a marker is clicked, weather camera live image as well as weather data is fetched (content extracting) and sanitized (content formatting). This data is passed to ImageViewer component (mashup creation). Qt Weather Cameras implements the most of the user interface of the application, i.e. rendering.

ImageViewer. ImageViewer works as a simple user interface component and displays the weather camera image as well as weather information. Implementing ImageViewer as a separate object provides support for reusing the code easily in other contexts.

ErrorManager. ErrorManager supervises both QtWeatherCameras and ImageViewer. Because of the Finnish Road Administration does not offer a proper API to fetch road weather information, it is necessary to extract this data from a

HTML document. This is prone to errors and therefore, the ErrorManager object is crucial to maintain the mashup functionality. Currently it simply reports the errors noticed to mashup developers.

6 Conclusions

Software mashups – software applications that are based on the ability to combine code and content from multiple web sites all over the world – have given rise to an entirely new way of creating software. It is increasingly common to find compelling web applications that aggregate and deliver images, text, and other data from numerous web sites in an innovative and often entirely unforeseen fashion.

Despite their wide deployment, the development of mashups is widely regarded as ad-hoc activity, and there are few approaches that support their development. Furthermore, approaches that do exist are most commonly tool oriented and do not take into account software engineering principles, such as modularity, decomposition, and modifiability.

In this paper, we have given a reference architecture for describing mashup applications. The goal of the architecture is to support the development of mashups in a fashion where they are created in a programmatic fashion. Furthermore, even if the actual design is different, we believe that the reference architecture gives vocabulary for addressing the different concerns of mashup development, similarly to architectural styles [13].

In the future, we hope to be able to generalize mashup development into a methodology where mashupping principles would be used as first-class citizens for software development. So far, most mashups have been built around data and other static components such as images that have been downloaded from different sources. The future challenge is to go beyond data, and to enable the flexible combination of software components from all over the world. Possible examples of such components include UI components, specialized financial libraries, mathematical libraries, and so on. Applications building on such components, referred to mashware in [15, 9], would finally unleash the full power of web-based development as well as the potential of the Web as the ultimate distribution system.

References

1. Crockford, D.: JavaScript: The Good Parts. O'Reilly Media (2008)
2. ECMA International, ECMAScript Language Specification ECMA-262. ISO/IEC 16262 (1998)
3. Fielding, R.T., Taylor, R.N.: Principled Design of the Modern Web Architecture. ACM Trans. Internet Technol. 2(2), 115–150 (2002)
4. Hartmann, B., Doorley, S., Klemmer, S.: Hacking, Mashing, Gluing: Understanding Opportunistic Design. IEEE Pervasive Computing 7(3), 46–54 (2008)
5. Lee, C., Tang, S., Tsai, C., Chen, Y.: Toward a New Paradigm: Mashup Patterns in Web 2.0. WSEAS Trans. Info. Sci. and App. 6(10), 1675–1686 (2009)
6. López, J., Pan, A., Bellas, F., Montoto, P.: Towards a Reference Architecture for Enterprise Mashups. Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos 2(2) (2008)

7. Mikkonen, T., Salminen, A.: Towards Pervasive Mashups in Embedded Devices. In: Proceedings of the 16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Macau, China, August 23-25, pp. 35–42. IEEE Computer Society (2010)
8. Mikkonen, T., Taivalsaari, A., Terho, M.: Lively for Qt: A platform for mobile web applications. In: Proc. 6th ACM Mobility Conference, Nice, France, September 2-4 (2009)
9. Mikkonen, T., Taivalsaari, A.: The Mashware Challenge: Bridging the Gap Between Web Development and Software Engineering. In: Proceedings of the 2010 Workshop on Future of Software Engineering Research, Santa Fe, New Mexico, USA, November 7-8, pp. 245–249. ACM Press (2010)
10. Nestler, T.: Towards a mashup-driven end-user programming of SOA-based applications. In: Proceedings of the 10th International Conference on Information Integration and Web-Based Applications & Services, Linz, Austria, November 24-26, pp. 551–554. ACM (2008)
11. Nyrhinen, F., Salminen, A., Mikkonen, T., Taivalsaari, A.: Lively mashups for mobile devices. In: Proceedings of the 1st International Conference on Mobile Computing, Applications and Services, San Diego, California, October 26-29 (2009)
12. Salminen, A., Nyrhinen, F., Mikkonen, T., Taivalsaari, A.: Developing Client-Side Mashups: Experiences, Guidelines and the Road Ahead. In: Proceedings of MindTrek 2010, Tampere, Finland, October 6-8. ACM (2010)
13. Shaw, M., Garlan, D.: Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall (1996)
14. Taivalsaari, A.: Mashware: The Future of Web Applications. Technical Report. UMI Order Number: SERIES13103. Sun Microsystems, Inc. (2009)
15. Taivalsaari, A., Mikkonen, T.: Mashups and Modularity: Towards Secure and Reusable Web Applications. In: Proceedings of 1st Workshop on Social Software Engineering and Applications, L'Aquila, Italy, September 16. IEEE Computer Society Press (2008)
16. Taivalsaari, A., Mikkonen, T.: Simplifying interactive programming with keywords 'that' and 'those'. In: Proceedings of the 36th Euromicro Conference on Software Engineering and Advanced Applications, Lille, France, September 1-3. IEEE Computer Society (2010)
17. Wong, J., Hong, J.: What do we "mashup" when we make mashups? In: Proceedings of the 4th International Workshop on End-User Software Engineering, Leipzig, Germany, May 12, pp. 35–39. ACM (2008)
18. Zang, N., Rosson, M., Nasser, V.: Mashups: who? what? why? In: Extended Abstracts on Human Factors in Computing Systems, CHI 2008, Florence, Italy, April 05-10, pp. 3171–3176. ACM, New York (2008)

Adaptation Patterns in Multi-Agent Architectures: The Gathering Pattern

J. Santiago Pérez-Sotelo[1], Carlos E. Cuesta[2], and Sascha Ossowski[1]

[1] Center for Intelligent Information Technologies (CETINIA),
[2] Department of Computer Languages and Systems II
Rey Juan Carlos University - Madrid, Spain
{josesantiago.perez,carlos.cuesta,
sascha.ossowski}@urjc.es

Abstract. Self-adaptation has been recently recognized as a basic architectural concern, required to deal with the growing complexity and the open nature of next-generation software systems. Multi-Agent Systems (MAS) describe self-aware and adaptable structures; their advanced approaches use *organizations* to provide further structuring, defining complex MAS architectures. The purpose of our work is to provide the basis for *adaptive organizations*, with an emphasis in the coordination mechanism, which is also adaptive. Our service-oriented agents gather in the context of predefined controls and protocols, creating an aggregate which can evolve using *adaptation patterns*. Eventually it would reach the stable form of an agreement. We provide a case study showing the interest of this approach, focusing specifically in the definition of adaptation patterns.

Keywords: Self-Adaptation, Adaptive Architecture, Multi-Agent Systems, Adaptation Pattern, Agreement Technologies, Service-Orientation.

1 Introduction

It is well known that in recent years the software systems have grown in complexity. This fact is leading software designers to rethink the strategy for handling it because today's difficulty goes beyond the administration of individual software. Many routine tasks are now being handled by systems themselves, including many actions related to the systems' own function. Complex systems are now able to observe themselves, and to adapt its structure and behaviour as necessary. These include not only functional behaviour, but also non-functional properties. Hence, considering self-adaptation as a basic architectural concern, there are many levels to take into account [5]. Some potential benefits include, for example, scalability to build system of systems, or abstraction to describe dynamic change, among others [6].

Concurrently, in Artificial Intelligence field, the Multi-Agent Systems (MAS) have been developed as a generic approach to solve complex problems. Some advanced approaches use the concept of *organizations* to provide further structuring. However, they still have limitations in order to reach actual self-adaptivity, i.e. having not only the capability to affect their settings, but also their own composition and types.

Our approach proposes a solution based in service-oriented MAS with Agreement Technologies [1][7][11] to deal with dynamism. The services are not offered directly

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 657–661, 2011.
© Springer-Verlag Berlin Heidelberg 2011

by agents, but through organizations. The goal is to provide *adaptive organizations*, with an emphasis in the *coordination* mechanism, which must also be adaptive, independently of the way in which the agents export their services. This paper focuses in *adaptation patterns*, which provide the first stage for this adaptivity.

The rest of this paper is structured as follows: section 2 describes the lifecycle of our self-organising structures, including adaptation patterns. The next section discusses this concept, using a case study (the *gathering* adaptation pattern) as a detailed example. Finally, conclusions and further lines of work are outlined.

2 Lifecycle of Self-organising Structures

Depending on concrete goals, any group of individuals can be arranged into certain structures by using two different kinds of mechanisms: *controls* and *protocols*. The former can be seen as elements which either *enforce* or *forbid* specific interactions (or architectural connections); and the latter, which are based on consensus and agreements, can either *enable* or *channel* behaviour. These elements will make possible to define the main inner structures in order to obtain agreement-based organizations. Once a primary structure is defined, an "elemental" group emerges as a preliminary organization, which will be referred as *an initiative* – it grows with the environmental dynamics; it is not yet fully established, but still evolving.

The previous paragraph implies three important concepts [8]: the *initiative*, a preliminary group of individuals which assemble with a certain structure; the *organization*, a established group, which is dynamically originated from an initiative (or statically defined *a priori*); and the *agreement*, which is the act by which an initiative became transformed into a "stable" organization. It can be seen as the consensus which is reached between individuals of the initial "seed" group.

Figure 1 summarizes briefly the *lifecycle* of our self-organizing structures [3]. This cycle can begin with a single *agent*, which is able to perform certain interactions and has the potential to export some *services*. Initially, it does not belong to any *organization* when reaches the system. However, it complies to a number of predefined *controls* and *protocols*, which "guide" the agent's interaction and enable it to maintain structured conversations with other agents, composing informal groups of agents.

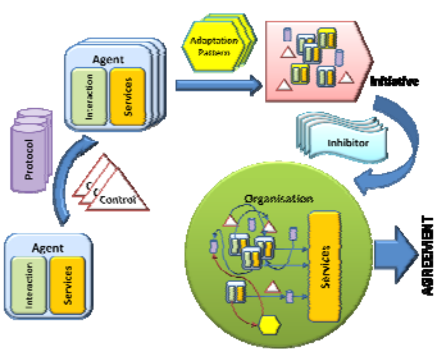


Fig. 1. Lifecycle of a Self-organizing Structure, from [3]

When an external change occurs, the system must react with an *adaptive* behaviour, and this is the functionality that must trigger the formation of our self-organizing structures (organizations). The system is provided with a number of *adaptation patterns* in order to achieve some desired reaction. These patterns are partial definitions of elements and relationships, which include enough information for an agent to *learn* how to perform some behaviour. Therefore, under the guidance of an adaptation pattern, certain agents within the group acquire specific functions, and begin to form an actual structure: this is what we call an *initiative*. Of course, these organizations are able to evolve themselves, and to participate in larger agreements [3].

3 Adaptation Patterns: The Gathering

As previously noted, an *initiative* can be generated from certain specific patterns, named *adaptation patterns*. Some of them have been previously identified, and receive such names as *Façade* or *Surveyor*, among others [2].

These are also *architectural patterns*; as already noted, the system is ultimately conceived as a *service-oriented architecture*; so methodologically, our first "stable" organizations must be conceived as the providers for certain high-level services. Then, these services will also be used as the starting point for the functional definition of our evolving organizations – even in the initiative stage.

As a case study we have identified another adaptation pattern: *Gathering*. In the following, we provide a template for this pattern, according to [9]. The template is similar in style and spirit to those used by [4], but some fields have been modified to address the dynamics of adaptive systems. Please note that several field descriptions are omitted or simplified due to space constraints.

The pattern describes the situation where several agents are located near a shared *venue*; and when some event happens at this place, these agents get capable to directly interact to each other – this initial event can trigger the creation of an *initiative*.

- Pattern name. **Gathering**
- Classification. Monitoring, Creational.
- Intent. Monitors the "space" containing computational elements (agents), routes monitoring data, allows the interaction and ultimately provides coordination.
- Context. [*Describes the context to apply the pattern.*] Gathering may be used when there is no initiative formed yet, i.e. when agents populate the "space" but they have not yet defined high-level services, providing an organisation.
- Motivation. The system needs to coordinate as a service ecosystem, where agents can enter or leave the environment at any time; this justifies why the behaviour cannot be completely pre-designed, and hence it must be emergent.
- Participants. [*Details elements in the pattern and lists their responsibilities*.] Gathering uses two (or more) agents and one venue. The underlying infrastructure must fulfil the role of the connector.

Like in other patterns, the template must be complemented with some model of the situation – in adaptation patterns, the *protocols* are of particular importance. For this purpose, we will use a specification in π-calculus [10]. We use this calculus due to its precision and expressive power, but it is *not* necessary for coding the patterns: several other mechanisms (or even formalisms) could be used instead.

This specification uses essentially standard polyadic π-calculus [10], with some convenient alterations. We use CSP notation instead of the original, and therefore we also reserve the asterisk symbol (∗) to represent the *replication* operation. We also assume a composite sequence (;) construct, and also the conditional (if A then B), which are conventionally encoded in a standard way [10]. We also add two simple functions: **near** (detects proximity) and **count**, which are provided for the algorithm's convenience. These could be encoded in the calculus, but could be provided in many different ways – so no more details are provided.

```
AgentX (sys_where) ::= …
        + (ν hi) sys_where!⟨hi⟩. sys_where?(where). τ. where?(event).
            (ν me, ret) ∗[ event!⟨me, ret⟩. ret?(you). τ. (me!⟨data⟩ | you!⟨info⟩) ];
              AgentX⟨sys_where⟩
        + …

Venue (sys_place) ::= …
        + (ν here, r) sys_place!⟨here, r⟩. τ.
           ∗[ r?(hi). if count(r) ≥ 2 then (ν event) here!⟨event⟩.
              ∗[ event?(ag_x, ragx). (
                ∗[ event?(ag_y, ragy). ragx!⟨ag_y⟩ ] | event!⟨ag_x, ragx⟩ ) ] ];
                     Venue ⟨sys_place⟩
        + …

SubSystem (sys_place, sys_where) ::= …
        +  sys_place?(here, r).
           ∗[ sys_where?(hi). if near(here, hi) then sys_where!⟨here⟩. r!⟨hi⟩) ];
               SubSystem⟨sys_place, sys_where⟩
        + …
```

The *AgentX* process describes the way in which a standard agent must behave in this pattern. He first awakes (*hi*) and asks the system where he is (*sys_where*). If he is close to a shared venue, then he receives its name (*where*). When something happens (*event*) there, the agent sends his own name (*me*) and a return channel (*ret*). He receives the name of some other agent (*you*) in this channel, so the two agents are now able to directly interact to each other – not using the venue anymore. This last part is a replicated sub-process (*[*A*]), so it repeats indefinitely and concurrently – meaning that the agent can communicate with many other agents, in pairs.

The *Venue* process describes the influence of the "place" on the agents. First, it informs the system of his existence (*here*) and provides a return channel (*r*) in which agents would eventually register. Once there are more two or more agents, the *event* is triggered and broadcast within a replicated process. The *event* channel is used to receive the name (*ag*_x) and return channel for every agent in the nearby, find a potential pair (*ag_y*) and communicate them, within a parallel process. Of course there are more sophisticated ways to define a broadcast – this is just an example.

And obviously, the *SubSystem* process acts as the connector – defines the behaviour of the part of the middleware which joins all these together.

As a final comment, it can be seen that every AgentX process is a *shifter* [3]: an "agent that changes" – he is altering his shape, i.e. his capability to interact to others. In the same context, the Venue defines a *changent*: an "agent of change" – he is the one who *inducts the change* on the rest of the elements in the pattern.

4 Conclusions and Further Work

This paper has explored structural concepts as the basis of an architectural approach to provide self-adaptivity to agent-based systems. Our concept of *initiative* can be considered as the starting point to provide mechanisms for changing composition patterns and element types within these systems. The required dynamism can be supported by an evolving architectural structure, based on combining predefined controls and protocols, in the context of the service-oriented, agent-based and organization-centric framework defined by Agreement Technologies (AT). Agents are *coordinated* and *reorganized* by their matching in *adaptation patterns*, and then evolve into “stable” organizations. The proposed patterns can be seen as the “seed” structures which trigger the (desired) behaviour, required to induct actual self-adaptivity.

These patterns define *protocols* which can be formally represented, as done in our example by using the π-calculus. Indeed, even when this approach seems already promising, these are just the first firm steps.

Further work will develop and implement variants of this approach in the AT-MAS framework, in order to refine them. The concepts are still evolving but even at this initial stage, the existing fragments of the approach have already proven its utility and expressive power. Our results suggest that adaptive architectures are indeed feasible, as the developed infrastructure can grow just by adding new adaptation patterns.

Acknowledgments. This work has been partially funded by Spanish Ministry of Science and Innovation through National Projects Agreement Technologies (CONSOLIDER CSD2007-0022), MULTIPLE (TIN2009-13838), and OVAMAH (TIN2009-13839-C03-02); and the European Union RTD Framework Programme, through COST Action Agreement Technologies (COST Action IC0801).

References

1. Agreement Technologies (AT) Project (2009), http://www.agreement-technologies.org/
2. Billhardt, H., et al.: Organisational Structures in Next-Generation Distributed Systems: Towards a Technology of Agreement. Multiagent and Grid Systems 7(2), 72–88 (2011)
3. Cuesta, C.E., Pérez-Sotelo, J.S., Ossowski, S.: Self-Organising Adaptive Structures: the Shifter Experience. ERCIM News 85, 35–36 (2011)
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional (1994)
5. Kephart, J.O., Chess, D.M.: The Vision for Autonomic Computing. IEEE Computer 36(1), 41–50 (2003)
6. Kramer, J., Magee, J.: Self-Managed Systems: an Architectural Challenge. In: Future of Software Engineering (FOSE@ ICSE 2007), pp. 259–268. IEEE (2007)
7. Ossowski, S.: Coordination in Multi-Agent Systems: Towards a Technology of Agreement. In: Bergmann, R., Lindemann, G., Kirn, S., Pěchouček, M. (eds.) MATES 2008. LNCS (LNAI), vol. 5244, pp. 2–12. Springer, Heidelberg (2008)
8. Pérez-Sotelo, J.S., Cuesta, C.E., Ossowski, S.: Towards Adaptive Service Ecosystems with Agreement Technologies. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 77–87. Springer, Heidelberg (2010)
9. Ramírez, A.J., Cheng, B.H.C.: Design Patterns for Developing Dynamically Adaptive Systems. In: SEAMS @ ICSE 2010, pp. 49–58. IEEE (2010)
10. Sangiorgi, D., Walker, D.: The Pi-calculus: a Theory of Mobile Processes. Cambridge University Press (2003)
11. Sierra, C., Botti, V.J., Ossowski, S.: Agreement Computing. KI 25(1), 57–61 (2011)

Author Index